Algorithms for Multiphase Partitioning

by

Matthew Jacobs

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)
in the University of Michigan
2017

Doctoral Committee:

Professor Selim Esedoḡlu, Co-Chair
Professor Jinho Baik, Co-Chair
Associate Professor Jason Corso
Professor Peter Miller

Matt Jacobs

majaco@umich.edu

ORCID iD: 0000-0001-9306-6495

For my parents.

ACKNOWLEDGMENTS

This thesis would not be possible without the contributions of many wonderful people. My friends and family, who patiently supported during my time in Michigan. My friend, Olivia Walch, who got me interested in the whole business of programming and computer vision. My collaborators, Pengbo Zhang and Ekaterina Merkurjev, who made collaborative research a joy. My advisor, Jinho Baik, who mentored and supported me over many years. I am especially thankful to my advisor and collaborator, Selim Esedoḡlu, for his mentorship, his insights and for accepting me as his student as I neared the end of the Ph.D. program. Finally, I am grateful for support from the National Science Foundation during part of this thesis work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PDE partial differential equation

SSL semi-supervised learning

GHC graph heat content

ABSTRACT

Algorithms for Multiphase Partitioning

by

Matthew Jacobs

Chair: Selim Esedoḡlu

Given a region D and a partition, $\mathbf{\Sigma}$, of D into a number of distinct phases $\mathbf{\Sigma} = (\Sigma_1, \ldots, \Sigma_N)$, a perimeter functional measures the area of the interfacial boundaries $\partial\Sigma_i \cap \partial\Sigma_j$ with respect to some measure on the surface normals. Perimeter functionals are at the heart of many important variational models, such as Mullins' model for grain boundary motion and the Mumford-Shah model for image segmentation. The gradient flow of perimeter functionals is a non-linear partial differential equation (PDE) known as curvature motion or curvature flow. Our focus is *threshold dynamics*, an efficient and elegant algorithm for simulating curvature flow. Recently, in [27], the authors re-derived and significantly generalized the threshold dynamics algorithm using a variational framework based on the heat content energy. The main thrust of this work is to further explore, analyze and extend threshold dynamics through the heat content energy. We use this framework to derive several new threshold dynamics schemes; namely "single growth" schemes which promise unconditional stability for virtually any situation of interest, and "auction" schemes which extend threshold dynamics to volume preserving curvature flow. Along the way, we answer an important and long standing question in the threshold dynamics community, and present applications to problems in machine learning.

CHAPTER 1

Introduction

Many important physical phenomena are described by a partitioning of space into distinct regions, often called phases. A prominent example occurs in polycrystalline materials, which includes most metals and ceramics. Each phase of a polycrystalline material is a grain (a single crystal piece) with a particular crystallographic structure. The behavior of such materials is largely determined by the configuration of the grains (their shapes and sizes), called the microstructure of the material. In fact, many of the bulk mechanical properties of the material are determined by the microstructure. The mismatch between the crystallographic orientations of neighboring grains induces an excess internal energy that is dissipated via reconfiguration of the microstructure. Naturally, the excess internal energy is proportional to the perimeter of the interfacial boundaries where distinct grains meet. If any of these materials are in a non-frozen state (e.g. if they are heated), the microstructure will often evolve by grain boundary motion, which in part acts to reduce the interfacial perimeters. Thus, these materials are intimately connected to minimal partition problems.

Minimal partition problems also arise in image processing and computer vision. A fundamental task in these fields is to partition a data set into some number of prescribed different classes. For example, one may want to segment an image into a number of distinct objects and regions. Here, a single phase in a partition may consist of pixels with homogenous color values, or small image patches with similar features. Obtaining high quality segmentations is closely tied to solving minimal partition problems. Indeed, object boundaries in natural images tend to be smooth curves or simple polygons.

Functionals measuring the perimeters of partitions are key to variational models for modeling and solving the above problems and phenomena. These include Mullins' model for grain boundary motion [66], and the Mumford-Shah model for image segmentation [67]. More recently, such variational models and their minimization via gradient descent have also been applied in the context of machine learning and artificial intelligence (e.g. graph partitioning models for supervised clustering of data [33]).

To minimize perimeter functionals, one naturally turns to gradient descent. In the variational setting, gradient descent is formulated as a time evolution PDE known as the Euler-Lagrange equation. The Euler-Lagrange equation for the L^2 gradient descent of the perimeter functional is motion by mean curvature of the interface boundaries. Given a point x in the interface, motion by mean curvature moves x in the direction normal to the interface with velocity proportional to the mean curvature of the interface at x. Thus, concave portions of the interface bulge outwards, while convex regions contract inwards.

Motion by mean curvature (also known as mean curvature flow) is a singular, degenerate, non-linear PDE. Mean curvature flow is best understood in the two phase case, which entails the evolution of single interface. In this setting, mean curvature flow is regularizing on short time scales. For example, C^2 curves and surfaces evolving under the flow instantaneously become smooth (in fact real analytic). However, the flow eventually develops singularities. Any convex curve or surface shrinks down and collapses to a single point in finite time. In two dimensions this is essentially the full story; Grayson showed that any closed embedded curve in two dimensions becomes convex and then collapses to a point under curvature flow [40]. In dimensions three and above, much more complexity is possible. In particular, the flows of certain embedded surfaces develop non-trivial singularities that result in topological changes. For example, a dumbbell shaped surface with certain proportions (connect two spheres with a long thin cylinder) will split at the handle into two pieces before collapsing into two separate points.

Understanding how to continue the flow after non-trivial singularities develop is the most challenging aspect of curvature flow. In the two-phase case, the theory of viscosity solutions allows one to define a well-posed notion of a solution past the formation of singularities. However, there is no analogous theory in the multi-phase case. Furthermore, in the multi-phase case much more complicated topological changes are possible. As a result, simulating curvature flow (especially in the multi-phase case) requires sophisticated numerical methods. Indeed, the most straight-forward approach known as front tracking (i.e. explicitly parameterizing the interfaces and moving them with the correct normal speed), entails handling topological changes in painful and heuristic ways.

Fortunately, there has been great success simulating curvature flow with methods that represent the interfaces implicitly. One of the most famous examples is the level-set method of Osher and Sethian [71]. Implicit methods represent embedded curves and surfaces as the level sets of a function. One obtains the correct motion by repeatedly solving a short time evolution equation on the entire function (or in a thin layer near the relevant level set) and then locating the new position of the level set. The principal advantage is the effortless handling of topological changes. Indeed, in the level set method, topological

changes require no reparameterizations or "surgeries". The focus of this work will be *threshold dynamics*, a level set method specialized for simulating curvature flow.

Originally proposed by Merriman, Bence, and Osher (MBO) in [63, 62], threshold dynamics – also known as the MBO algorithm or diffusion generated motion – is a very efficient algorithm for approximating the motion by mean curvature of an interface. The algorithm generates a discrete in time approximation to mean curvature motion by alternating two very simple steps: convolution with a kernel, and pointwise thresholding. Among its benefits are 1. implicit representation of the interface as in the phase field or level set methods, allowing for graceful handling of topological changes, 2. unconditional stability (for certain kernels), where the time step size is restricted only by accuracy considerations, and 3. very low per time step cost when implemented on uniform grids.

Recently, Esedoḡlu and Otto gave a variational interpretation of the MBO scheme based on the heat content energy [27]. Given a partition, the heat content measures the amount of heat that escapes from each phase under a short time diffusion generated by a convolution kernel K. At small time scales, the amount of heat that escapes is proportional to the perimeter of the boundary. Thus, the heat content energy is a non-local approximation to the perimeter functional [1], [64], and thus is naturally connected to curvature flow. Esedoḡlu and Otto showed that one may recover and generalize the MBO algorithm by successively minimizing linearizations of the heat content. This procedure shows that for certain kernels, the heat content is dissipated at every step of the MBO algorithm. Thus, the heat content is a Lyapunov functional for the MBO scheme, implying unconditional stability of the algorithm.

The variational interpretation of threshold dynamics via the heat content energy is at the heart of this work. The starting point of our investigations is the theory developed in [27] and [25]. We build upon this theory, using the heat content to derive and analyze new "single growth" variants of threshold dynamics. A natural outgrowth of our schemes are new kernel constructions, which allows us to answer some long standing questions in the threshold dynamics community. In addition, we derive a new multi-phase volume preserving MBO algorithm and we discuss and present applications to machine learning, in particular the semi-supervised learning (SSL) problem.

1.1 Background and Previous Work

We now develop some notation, background and previous work that will be relevant for all that follows. We will be concerned with possibly anisotropic interfacial energies defined on partitions of a domain D. In Chapters 2-4, D will be the d-dimensional torus, i.e. a cube in

$\mathbb{R}^d$ with periodic boundary conditions. In Chapter 5, D will be a weighted graph $(\mathcal{V}, W)$, which is a finite collection of points $\mathcal{V}$ along with a symmetric matrix $W : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ describing the connection strength between any two points. By a partition of D, we mean a prescribed number N of sets $\Sigma_1, \ldots, \Sigma_N \subseteq D$ that satisfy

$$\bigcup_{i=1}^{N} \Sigma_i = D \text{ and } \Sigma_i \cap \Sigma_j = (\partial\Sigma_i) \cap (\partial\Sigma_j) \text{ for } i \neq j \tag{1.1}$$

When working with two phase partitions we will simplify our notation. We will represent (Σ_1, Σ_2) as a single phase $\Sigma := \Sigma_1$, since we may recover Σ_2 by taking the complement.

Let H^s denote the s-dimensional Hausdorff measure on D. Then the perimeter functional of a two phase partition Σ is

$$E(\Sigma) = H^{d-1}(\partial\Sigma) = \int_{\partial\Sigma} dH^{d-1}(x). \tag{1.2}$$

and the perimeter functional for a multiphase partition $\mathbf{\Sigma} = (\Sigma_1, \ldots, \Sigma_N)$ is

$$E(\mathbf{\Sigma}) = \sum_{i=1}^{N} H^{d-1}(\partial\Sigma_i) = \sum_{i=1}^{N} \int_{\partial\Sigma_i} dH^{d-1}(x) \tag{1.3}$$

We will primarily be interested in more general surface energies where we allow the perimeter measure to be anisotropic. This level of generality is essential for material sciences applications, where interaction energies may depend on the misorientation angles of two grains and the normal of the grain boundary between them. Anisotropic surface measures are also useful in computer vision applications, for example if one wants to favor detecting certain shapes over others.

Given a surface tension σ, a strictly positive, continuous, even function $\sigma : \mathbb{S}^{d-1} \to (0, \infty)$ we consider the more general version of (1.2)

$$E(\Sigma, \sigma) = \int_{\partial\Sigma} \sigma(n(x))\, dH^{d-1}(x) \tag{1.4}$$

where $n(x)$ denotes the outward unit normal to $\partial\Sigma$. It will be convenient to assume that the surface tension σ has been extended to $\sigma : \mathbb{R}^d \to \mathbb{R}^+$ as

$$\sigma(x) = |x|\sigma\left(\frac{x}{|x|}\right) \text{ for } x \neq 0 \tag{1.5}$$

so that it is positively 1-homogeneous. We will assume that σ is then a convex function

on $\mathbb{R}^d$; this condition will ensure well-posedness (lower semi-continuity) of the two-phase energy (1.4). Note that the above conditions imply that σ is a norm on $\mathbb{R}^d$. Define the unit ball (i.e. the *Frank diagram)* B_σ *of* σ *as*

$$B_\sigma = \{x \,:\, \sigma(x) \leq 1\}$$

which is thus a closed, convex, centrally symmetric set. We will further require B_σ to be smooth and strongly convex; this implies that we stay clear of the crystalline cases (where B_σ is a polytope) except via approximation. In two dimensions, we will also write $\sigma = \sigma(\theta)$, where θ is the angle that the unit normal makes with the positive x_1-axis. In that case, strong convexity of B_σ is equivalent to the condition

$$\sigma''(\theta) + \sigma(\theta) > 0.$$

The *Wulff shape* W_σ associated with the anisotropy σ is defined as

$$W_\sigma = \left\{ y \,:\, \sup_{x \in B_\sigma} x \cdot y \leq 1 \right\}.$$

The sets B_σ can in turn be obtained from W_σ by the formula

$$B_\sigma = \left\{ x \,:\, \sup_{y \in W_\sigma} x \cdot y \leq 1 \right\},$$

exhibiting the well known duality between B_σ and W_σ. Our assumptions on B_σ imply that W_σ is also strongly convex and has smooth boundary.

We will also consider the multiphase extension of energy (1.4) to partitions. Let $N \in \mathbb{N}^+$ denote the number of phases, and define the set of pairs of indices:

$$\mathcal{I}_N = \{(i,j) \in \{1,\dots,N\} \times \{1,\dots,N\} : i \neq j\}, \tag{1.6}$$

which doubly enumerates the set of distinct interfaces $\partial\Sigma_i \cap \partial\Sigma_j$. To each interface we associate a potentially different surface tension $\sigma_{i,j} = \sigma_{j,i}$ satisfying the same properties as the two phase surface tension σ above. With these definitions, our multiphase energy is:

$$E(\boldsymbol{\Sigma}, \boldsymbol{\sigma}) = \sum_{(i,j) \in \mathcal{I}_N} \int_{(\partial\Sigma_i) \cap (\partial\Sigma_j)} \sigma_{i,j}(n(x))\, dH^{d-1}(x) \tag{1.7}$$

where $\boldsymbol{\sigma} = \{\sigma_{ij}\}_{(i,j) \in \mathcal{I}_N}$ denotes the collection of surface tensions. Note that in the multiphase case, even if the surface tensions are isotropic (i.e. constant functions), energy (1.7)

does not simplify to (1.3) – each interface may have a different constant.

For $d = 2$ or 3, we will study approximations for L^2 gradient flow of energies (1.4) and (1.7), which are known as weighted mean curvature flow (of an interface and a network). The normal speed of a given interface with surface tension σ in three dimensions under this flow is

$$v_\perp(x) = \mu(n(x)) \left(\left(\partial_{s_1}^2 \sigma(n(x)) + \sigma(n(x)) \right) \kappa_1(x) + \left(\partial_{s_2}^2 \sigma(n(x)) + \sigma(n(x)) \right) \kappa_2(x) \right) \tag{1.8}$$

where κ_1 and κ_2 are the two principal curvatures, and ∂_{s_i} denotes differentiation along the great circle on $\mathbb{S}^2$ that passes through $n(x)$ and has as its tangent the i-th principal curvature direction. The function $\mu : S^2 \to \mathbb{R}^+$ is known as the mobility factor, which determines the speed of the flow in each normal direction (as with the surface tensions, in the multiphase case each interface may have a different mobility). In two dimensions, the expression simplifies to

$$v_\perp(x) = \mu(n(x))(\sigma(n(x)) + \sigma''(n(x)))\kappa(x), \tag{1.9}$$

where $\kappa(x)$ is the curvature at x.

While materials science literature e.g. [23, 46] appears to allow the mobility factor $\mu : \mathbb{S}^{d-1} \to \mathbb{R}^+$ in (1.8) or (1.9) to be any positive function of the normal, a natural and important subclass of mobilities are those μ that have a *convex* one-homogeneous extension (as in (1.5)) to $\mathbb{R}^d$. Indeed, as explained in [7], motion law (1.8) arises as gradient descent for energy (1.4) with respect to a *norm* $\mu : \mathbb{R}^d \to \mathbb{R}$ on normal vector fields on $\partial\Sigma$ e.g. via the well-known discrete-in-time *minimizing movements* [22] procedure of Almgren, Taylor & Wang [2], and independently, Luckhaus & Sturzenhecker [55]:

$$\Sigma^{k+1} = \arg\min_{\Sigma} \left\{ E(\Sigma, \sigma) + \frac{1}{\delta t} \int_{\Sigma \triangle \Sigma^k} d_{\Sigma^k}^{\mu^*}(x)\, dx \right\} \tag{1.10}$$

where $d_{\Sigma^k}^{\mu^*}$ denotes the *distance function* to the interface $\partial\Sigma^k$ at the k-th time step, with respect to the *dual norm* μ^* of the norm μ:

$$\mu^*(x) = \sup_{y\,:\,\mu(y)\leq 1} x \cdot y.$$

In addition to (1.8), a condition known as the *Herring angle condition* [44] holds along triple junctions: For $d = 3$, at a junction formed by the meeting of the three phases Σ_i, Σ_j,

and Σ_k , this condition reads

$$(\ell \times n_{i,j})\sigma_{i,j}(n_{i,j}) + (\ell \times n_{j,k})\sigma_{j,k}(n_{j,k}) + (\ell \times n_{k,i})\sigma_{k,i}(n_{k,i}) \\ + n_{j,i}\sigma'_{i,j}(n_{i,j}) + n_{k,j}\sigma'_{j,k}(n_{j,k}) + n_{i,k}\sigma'_{k,i}(n_{k,i}) = 0 \quad (1.11)$$

where $n_{i,j}$ is the unit normal vector to the interface $(\partial\Sigma_i) \cap (\partial\Sigma_j)$ pointing in the Σ_i to Σ_j direction, $\ell = n_{j,k} \times n_{i,j}$ is a unit vector tangent to the triple junction, and $\sigma'_{i,j}(n_{i,j})$ denotes derivative of $\sigma_{i,j}$ taken on $\mathbb{S}^2$ in the direction of the vector $\ell \times n_{i,j}$. In the isotropic setting, (1.11) simplifies to the following more familiar form, known as Young's law:

$$\sigma_{i,j}n_{i,j} + \sigma_{j,k}n_{j,k} + \sigma_{k,i}n_{k,i} = 0. \quad (1.12)$$

Finally, we note that well-posedness (lower semi-continuity) of the multiphase energy (1.7) in its full generality is a complicated matter [3]. At the very least, the surface tensions $\sigma_{i,j} : \mathbb{R}^d \to \mathbb{R}^+$ need to be convex and satisfy a pointwise triangle inequality

$$\sigma_{i,j}(n) + \sigma_{j,k}(n) \geq \sigma_{i,k}(n) \quad (1.13)$$

for all distinct i, j, and k, and all $n \in \mathbb{S}^{d-1}$. In case the $\sigma_{i,j}$ are positive constants, (1.13) is known to be also sufficient for well-posedness of model (1.7) [65].

1.1.1 Threshold dynamics

In its simplest form, the two-phase MBO algorithm as presented in the original paper [62] generates a discrete in time, continuous in space approximation to the motion by mean curvature of an interface $\partial\Sigma^0$ (given as the boundary of an initial set Σ^0) as follows:

Algorithm 1.1: (MBO'92 [62])

Given a time step size $\delta t > 0$, alternate the following steps:

1. Convolution:
$$\psi^k = K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma^k} \quad (1.14)$$

2. Thresholding:
$$\Sigma^{k+1} = \left\{ x \,:\, \psi^k(x) \geq \frac{1}{2} \right\}. \quad (1.15)$$

where Σ^k denotes the approximate solution at time $t = k\delta t$, and the convolution kernel

$K \in L^1(\mathbb{R}^d)$ satisfies

$$K(x) \in L^1(\mathbb{R}^d) \ , \ xK(x) \in L^1(\mathbb{R}^d), \text{ and } K(x) = K(-x) \tag{1.16}$$

together with

$$\int_{\mathbb{R}^d} K(x)\, dx > 0. \tag{1.17}$$

For convenience, we will write

$$K_\varepsilon(x) = \frac{1}{\varepsilon^d} K\left(\frac{x}{\varepsilon}\right)$$

for the rescaled versions of a given convolution kernel K. In the original papers [63, 62], the kernel K is taken to be the Gaussian:

$$G(x) = \frac{1}{(4\pi)^{\frac{d}{2}}} \exp\left(-\frac{|x|^2}{4}\right)$$

although the possibility of choosing it to be not necessarily radially symmetric for anisotropic curvature motions is also mentioned.

There have been multiple studies devoted to the question of convergence for Algorithm 1.1. In [58], [74], [73], consistency of the scheme is studied via Taylor expansion after one step of the algorithm is applied on a set with a smooth boundary. Rigorous convergence results were first given in [30] and [4]. [48] studies the algorithm with fairly general convolution kernels K, and establishes its convergence to the viscosity solution [31, 20] of certain anisotropic curvature flows provided that K satisfies certain conditions, chief among which is pointwise positivity. Positivity of K implies that the scheme preserves a comparison principle known to hold for the evolution (1.8) and is crucial in the viscosity solutions approach.

The inverse question of constructing a kernel K to induce particular normal velocities using Algorithm 1.1 has also been considered in the literature. In Chapter 3, we provide a complete and explicit answer to this question. We will postpone reviewing the relevant literature until Section 3.3 in Chapter 3.

1.1.2 The heat content energy

In [27], a variational formulation for the original MBO scheme, Algorithm 1.1, was given. In particular, it was shown that the following functional, the heat content energy, defined on sets, with kernel K chosen to be the Gaussian G, which had previously been established

[1, 64] to be a non-local approximation to (isotropic) perimeter, is dissipated by the MBO scheme at every step, regardless of time step size:

$$\mathrm{HC}_{\sqrt{\delta t}}(\Sigma, K_{\sqrt{\delta t}}) = \frac{1}{\sqrt{\delta t}} \int_{\Sigma^c} K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma}\, dx. \tag{1.18}$$

Thus, the heat content energy is a Lyapunov functional for Algorithm 1.1, establishing its unconditional gradient stability. Moreover, the following *minimizing movements* [2, 55] interpretation involving (1.18) for Algorithm 1.1 was given in [27]:

$$\Sigma^{k+1} = \arg\min_{\Sigma} \mathrm{HC}_{\sqrt{\delta t}}(\Sigma, K_{\sqrt{\delta t}}) + \frac{1}{\sqrt{\delta t}} \int (\mathbf{1}_{\Sigma} - \mathbf{1}_{\Sigma^k}) K_{\sqrt{\delta t}} * (\mathbf{1}_{\Sigma} - \mathbf{1}_{\Sigma^k})\, dx \tag{1.19}$$

where the kernel K was again taken to be the Gaussian.

Let us recall the following fact from [27] that ensures (1.18) is a Lyapunov functional for Algorithm 1.1, establishing the connection between the variational problem (1.18) and threshold dynamics, and underlining the significance of the Fourier transform of the kernel $\widehat{K}$:

Proposition 1.1.1. *(from [27]) Let K satisfy (1.16) and (1.17). If $\widehat{K} \geq 0$, threshold dynamics algorithm, Algorithm 1.1, decreases the heat content energy (1.18) at every time step, regardless of the time step size.*

In [27], the variational formulation (1.19) was then extended to the multiphase energy (1.7) in case the surface tensions $\sigma_{i,j}$ are constant but possibly distinct:

$$E(\mathbf{\Sigma}, \boldsymbol{\sigma}) = \sum_{(i,j)\in\mathcal{I}_N} \sigma_{i,j} H^{d-1}(\partial\Sigma_i \cap \partial\Sigma_j) \tag{1.20}$$

in which case the heat content energy becomes

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}, K_{\sqrt{\delta t}}, \boldsymbol{\sigma}) = \frac{1}{\sqrt{\delta t}} \sum_{(i,j)\in\mathcal{I}_N} \sigma_{i,j} \int_{\Sigma_j} K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_i}\, dx. \tag{1.21}$$

We also consider a relaxation of (1.21):

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{u}, K_{\sqrt{\delta t}}, \boldsymbol{\sigma}) = \frac{1}{\sqrt{\delta t}} \sum_{(i,j)\in\mathcal{I}_N} \sigma_{i,j} \int_{D} u_j K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_i}\, dx \tag{1.22}$$

over the following convex set of functions satisfying a box constraint:

$$\mathcal{K} = \left\{ \mathbf{u} \in L^1(D, [0,1]^N) \, : \, \sum_{i=1}^N u_i(x) = 1 \text{ a.e. } x \in D \right\}. \tag{1.23}$$

There is a corresponding minimizing movements scheme that can be derived from (2.27) that leads to the following extension of threshold dynamics, Algorithm 1.2, to the constant but possibly unequal surface tension multiphase energy (1.20).

Algorithm 1.2: ([27])

Given a time step size $\delta t > 0$, alternate the following steps:

1. Convolution:
$$\psi_i^k = K_{\sqrt{\delta t}} * \sum_{j \neq i} \sigma_{i,j} \mathbf{1}_{\Sigma_j^k}. \tag{1.24}$$

2. Thresholding:
$$\Sigma_i^{k+1} = \left\{ x \, : \, \psi_i^k(x) \leq \min_{j \neq i} \psi_j^k(x) \right\}. \tag{1.25}$$

Various conditions are provided in [27] for ensuring that Algorithm 1.2 is unconditionally gradient stable (decreases energy (1.21) for any $\delta t > 0$). The question turns out to be interesting, with connections to isometric embeddability of finite metric spaces into Euclidean spaces. In particular, the triangle inequality (1.13) on $\sigma_{i,j}$ appears to be neither necessary nor sufficient.

Turning to anisotropy, i.e. normal dependent surface tensions $\sigma = \sigma(n)$ and the more general convolution kernels it requires, we recall the following facts from [25]:

Proposition 1.1.2. *(from [25]) Let Σ be a compact subset of $\mathbb{R}^d$ with smooth boundary. Let $K : \mathbb{R}^d \to \mathbb{R}$ be a kernel satisfying (1.16). Then:*

$$\lim_{\delta t \to 0^+} \boldsymbol{HC}_{\sqrt{\delta t}}(\Sigma, K_{\sqrt{\delta t}}) = \int_{\partial \Sigma} \sigma_K\big(n(x)\big) \, dH^{d-1}(x)$$

where the surface tension $\sigma_K : \mathbb{R}^d \to \mathbb{R}^+$ is defined as

$$\sigma_K(n) := \frac{1}{2} \int_{\mathbb{R}^d} |n \cdot x| K(x) \, dx. \tag{1.26}$$

In Chapter 2 we will in fact prove a stronger, Gamma convergence version of Proposition 1.1.2 for a class of kernels that include sign changing ones. Let us also note the following *Barrier Theorem* from [25] that places a restriction on the positivity of convolution kernels in terms of the Wulff shape W_σ of the given anisotropy σ.

Figure 1.1: Some examples of non-zonoidal polytopes [25]. Note that each polytope has at least one face that is not centrally symmetric.

Theorem 1.1.3. *(from [25]) Threshold dynamics algorithm (1.14) & (1.15) with a positive kernel K can approximate weighted mean curvature flow (1.8) associated with an anisotropic surface tension $\sigma : \mathbf{S}^{d-1} \to \mathbb{R}$ (for some choice of mobility $\mu : \mathbb{S}^{d-1} \to \mathbb{R}$) if and only if the corresponding Wulff shape W_σ is a* zonoid.

Zonoids are centrally symmetric convex bodies that are limits, in the Hausdorff topology, of *zonotopes*. Zonotopes are convex bodies whose support functions have the form

$$h_A(x) = \sum_{i=1}^{N} m_i |x \cdot v_i| \tag{1.27}$$

for some collection of positive constants m_i and unit vectors v_i. A simple example of a zonotope is a hypercube of any dimension. In $\mathbb{R}^d$, a convex polytope with nonempty interior is a zonotope if and only if every $d-1$ dimensional face of it is a zonotope. Thus, for $d = 2$, any centrally symmetric, convex body is a zonoid. For $d = 3$ and higher, this is no longer the case. A simple example of a non-zonoid in $\mathbb{R}^3$ is the octahedron– the triangular faces of the octahedron are not centrally symmetric (see Figure 1.1). Moreover, there exists a neighborhood of the octahedron that contains no zonoids. In fact, the space of zonoids is nowhere dense in the space of convex bodies for dimensions $d \geq 3$. Theorem 1.1.3 implies that there is no monotone threshold dynamics scheme for an anisotropy σ the Wulff shape W_σ of which is non-zonoidal, even though W_σ may be smooth and strictly convex. See [39, 14] for these facts and much more information about zonoids.

1.2 Summary of Contributions

Finally, we conclude this introductory chapter with a summary of the original contributions contained in this thesis.

- Chapter 2
  - New "single growth" variants of the MBO scheme which guarantee unconditional stability for an extremely wide class of convolution kernels. Combined with the kernel constructions in Chapter 3, this implies that the multi-phase single growth algorithm is unconditionally stable for essentially any network of surface tensions and mobilities. This full level of generality is a first for threshold dynamics schemes.
  - In the two-phase case, we prove that our single growth scheme with a nonnegative convolution kernel converges to the viscosity solution of the induced weighted curvature flow.
  - An expansion of the Gamma-convergence proof in [27] for the heat content energy. Our new Gamma-convergence result allows for non-radially symmetric kernels and certain sign-changing kernels. The Gamma-convergence result gives us hope that the energy based convergence arguments in [54] will extend to the kernels covered by our argument. This is of particular interest as the viscosity solution based convergence arguments of [48] do not apply to sign-changing kernels. Unfortunately, the class of sign-changing kernels that are admissible for our argument does not include kernels inducing non-zonoidal anisotropies. This is perhaps the white whale of this thesis.
- Chapter 3
  - A complete characterization of anisotropy-mobility pairs which may be induced under threshold dynamics with a nonnegative kernel. For admissible pairs, we give a completely explicit kernel construction in 2 dimensions, and an explicit construction modulo the inversion of a Radon transform in 3 dimensions. The resulting kernels are smooth and have compact support. This answers an important and long-standing question in the threshold dynamics community.
  - An explicit Fourier domain kernel construction for any smooth anisotropy and mobility pair in any dimension. The resulting kernel is Schwartz class and nonnegative in the Fourier domain. This improves upon the kernel construction in [15], which was restricted to the (important) special case where the anisotropy

and mobility are equal and had a singularity at the origin for non-ellipsoidal anisotropies.

- Chapter 4
  - Auction dynamics: a novel variant of the MBO scheme for volume preserving curvature flow. Strict volume constraints on each phase transform the thresholding step of the classic MBO scheme into an instance of the assignment problem. We efficiently solve this step using auction algorithms [9]. The principal advantage of our approach is the partition and the Lagrange multipliers corresponding to the volume constraints are updated in the same step. This ensures that the volume constraints are always exactly satisfied, and allows for use of the algorithm in situations where the interfacial boundaries are extremely rough or irregular. This is especially important for partitioning problems on graphs, where boundaries and interiors are poorly defined. In addition, we provide a variant of the algorithm where the volume of each phase is allowed to vary between certain user provided upper and lower bounds, and explain how to incorporate random fluctuations due to temperature.
- Chapter 5
  - We introduce the graph analogue of the heat content energy, the graph heat content (GHC) energy. GHC is closely related to the weighted graph cut, and as a result we can solve semi-supervised graph partitioning problems by minimizing GHC. We introduce several new graph MBO algorithms for the SSL problem, based on our results in Chapters 2 and 4. Experimental results on benchmark machine learning datasets show that our methods are faster and maintain accuracy at lower fidelity percentages than other state-of-the-art variational methods for the SSL problem.

CHAPTER 2

Convolution Kernels and Stability of Threshold Dynamics Methods [1]

2.1 Introduction

The elegant, streamlined nature of threshold dynamics has made it amenable to analysis and the focus of a number of theoretical investigations, see e.g. [48, 15, 27, 54, 25] and references therein. The various consistency, stability, and convergence statements contained in these contributions require various assumptions on the kernel used in the convolution step of the algorithm, such as positivity in the physical or the Fourier domain. In this chapter, we present a number of new results on the original threshold dynamics algorithm and some of its variants and extensions that significantly enlarge the class of admissible kernels. We also demonstrate that some of the remaining restrictions are necessary. There are multiple reasons for seeking an extension of the theory to more general kernels. Three of these are:

1. The barrier theorem, Theorem 1.1.3, from [25] implies that certain anisotropic surface tensions can only be generated using sign changing kernels.

2. Enlarging the class of admissible kernels simplifies kernel constructions (see Chapter 3).

3. In applications such as graph partitioning, there is often little control on the properties of the convolution kernel that is typically constructed from the given edge weights of the graph.

[1] Joint work with Selim Esedoḡlu [26]. Submitted to *SIAM Journal on Numerical Analysis*.

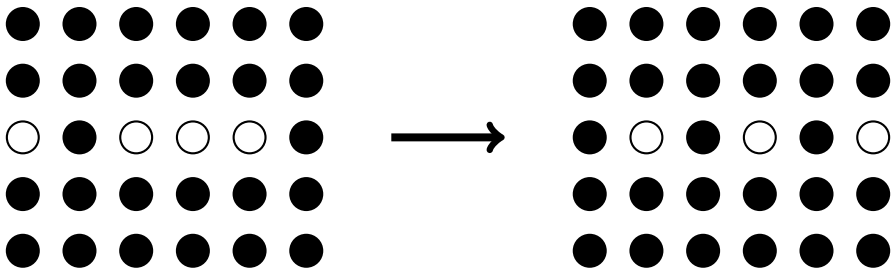

Figure 2.1: One step of the standard algorithm under the action of kernel K_1 on the periodic lattice $\mathbb{Z}/6\mathbb{Z} \bigoplus \mathbb{Z}/5\mathbb{Z}$. The updated configuration has a higher energy than the starting configuration.

2.2 New Variants of Threshold Dynamics

Here, we will consider extensions of the basic MBO algorithm (1.14) & (1.15) that allow us to dispense with various requirements on the convolution kernel K and, in the multiphase setting, on the surface tensions σ.

2.2.1 Non-monotone energy and oscillating solutions

We first establish a partial converse to Proposition 1.1.1, showing that assumption $\widehat{K} \geq 0$ on the Fourier transform of the kernel is not spurious.

Example 1:

Let the convolution kernel be given by

$$K_1\big((x_1, x_2)\big) = \begin{cases} 1/3 & (x_1, x_2) = (\pm 1, 0) \\ 1/9 & (x_1, x_2) = (0, \pm 1) \text{ or } (0, 0) \\ 0 & \text{otherwise} \end{cases}$$

Then the Fourier transform of K_1 must change signs since the origin is not the global maximizer of K_1. Note that K_1 is not an artificial example, K_1 has a similar structure to some of the nonnegative kernels constructed in Chapter 3 (see Figure 3.3). Figure 2.1 shows an example where a step of the algorithm with K_1 increases the heat content energy. The right hand side configuration has 6 broken horizontal bonds and 6 broken vertical bonds, while the left hand side has 4 broken horizontal bonds and 8 broken vertical bonds. K_1 assigns horizontal bonds a strength of 1/3 and vertical bonds a strength of 1/9 thus if we compare the two energies we see that $E_{RHS} - E_{LHS} = 6*1/3+6*1/9-(4*1/3+8*1/9) = 8/9$ thus the energy has increased under the algorithm.

Example 2:

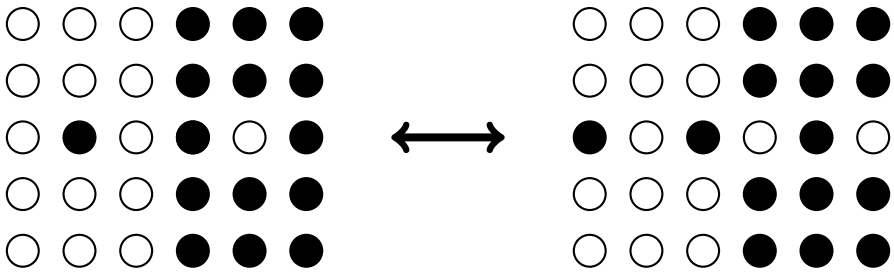

Figure 2.2: Behavior of the standard algorithm under the action of kernel K_1 on the periodic lattice $\mathbb{Z}/6\mathbb{Z} \bigoplus \mathbb{Z}/5\mathbb{Z}$. The algorithm gets trapped in a periodic loop between the two configurations above. The configuration on the right has a higher energy

In fact it is possible for the algorithm to get trapped in a periodic cycle where one of the configurations in the cycle has a higher energy than the others. Figure 2.2 shows an example of the algorithm with K_1 getting trapped in a 2-cycle where the right hand configuration has a higher energy. Both configurations have the same number of broken horizontal bonds. However the left hand side has 4 broken vertical bonds while the right hand side has 8 broken vertical bonds.

2.2.2 A new variant: Single growth

Next, we show how to modify the original two-phase MBO algorithm, Algorithm 1.1, to a slightly more costly version (entailing two convolutions per time step as opposed to one) so that the assumption $\widehat{K} \geq 0$ can be dramatically relaxed while maintaining the energy dissipation property.

Proposition 2.2.1. *If the convolution kernel K is of the form $K = K^1 + K^2$ with $K^1 \geq 0$ and $\widehat{K^2} \geq 0$, then Algorithm 2.1 dissipates the two-phase heat content energy at every step. Furthermore if K^1 is positive in a neighborhood of the origin or $\hat{K}^2(0)$ is positive then if an iteration of Algorithm 1.1 changes a configuration Σ by a set of positive measure, an iteration of Algorithm 2.1 also changes Σ by a set of positive measure and strictly decreases the energy.*

Proof. We will show that energy (1.18) is dissipated going from Σ^k to $\Sigma^{k+\frac{1}{2}}$. The argument going from $\Sigma^{k+\frac{1}{2}}$ to Σ^{k+1} is the same, we simply apply what follows to the complements of the sets instead. Let $\varphi = \mathbf{1}_{\Sigma^{k+\frac{1}{2}}} - \mathbf{1}_{\Sigma^k}$. Then $\varphi(x)$ is pointwise nonnegative since $\Sigma^k \subset \Sigma^{k+\frac{1}{2}}$. Comparing the energies we have:

$$\begin{aligned}
&\mathrm{HC}_{\sqrt{\delta t}}(\Sigma^{k+\frac{1}{2}}, K_{\sqrt{\delta t}}) - \mathrm{HC}_{\sqrt{\delta t}}(\Sigma^k, K_{\sqrt{\delta t}}) \\
&\qquad = \frac{1}{\sqrt{\delta t}}\Big(\int_D \varphi(x)(K_{\sqrt{\delta t}} * (\mathbf{1}_{(\Sigma^k)^c} - \mathbf{1}_{\Sigma^k}))(x)\, dx - \int_D \varphi(x)(K_{\sqrt{\delta t}} * \varphi)(x)\, dx \Big).
\end{aligned}$$

Algorithm 2.1: Two phase single growth

Alternate the following steps:

1. 1st Convolution:
$$\psi^{k+\frac{1}{2}} = K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma^k} \tag{2.1}$$

2. 1st Thresholding:
$$\Sigma^{k+\frac{1}{2}} = \Sigma^k \cup \left\{ x \,:\, \psi^{k+\frac{1}{2}}(x) \geq \frac{1}{2} \right\}. \tag{2.2}$$

3. 2nd Convolution:
$$\psi^{k+1} = K_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma^{k+\frac{1}{2}}} \tag{2.3}$$

4. 2nd Thresholding:
$$\Sigma^{k+1} = \Sigma^{k+\frac{1}{2}} \setminus \left\{ x \,:\, \psi^{k+1}(x) \leq \frac{1}{2} \right\}. \tag{2.4}$$

Note that $(K_{\sqrt{\delta t}} * (\mathbf{1}_{(\Sigma^k)^c} - \mathbf{1}_{\Sigma^k}))(x) > 0$ if and only if $\psi^{k+\frac{1}{2}}(x) < \frac{1}{2}$ and thus if and only if $\varphi(x) = 0$. Therefore

$$\frac{1}{\sqrt{\delta t}} \int_D \varphi(x)(K_{\sqrt{\delta t}} * (\mathbf{1}_{(\Sigma^k)^c} - \mathbf{1}_{\Sigma^k}))(x)\, dx \leq 0.$$

To establish the dissipation of energy, it remains to show that

$$-\frac{1}{\sqrt{\delta t}} \int_D \varphi(x)(K_{\sqrt{\delta t}} * \varphi)(x)\, dx \leq 0.$$

Let L be the periodic lattice associated to D. Then using the Fourier series expansion we have

$$-\int_D \varphi(x)(K_{\sqrt{\delta t}} * \varphi)(x)\, dx = -\sum_{\alpha \in L} \widehat{\varphi}(\alpha)^2 \widehat{K}(\alpha\sqrt{\delta t}). \tag{2.5}$$

If K is nonnegative then it is clear that the left hand side of the above equation is ≤ 0 and if $\widehat{K}$ is nonnegative then is clear that the right hand side is ≤ 0. Therefore if K can be split into a sum $K = K^1 + K^2$ where $K^1 \geq 0$ and $\widehat{K^2} \geq 0$ we have

$$\mathrm{HC}_{\sqrt{\delta t}}(\Sigma^{k+\frac{1}{2}}, K_{\sqrt{\delta t}}) - \mathrm{HC}_{\sqrt{\delta t}}(\Sigma^k, K_{\sqrt{\delta t}}) \leq 0.$$

Now we prove the second statement. By the above it is enough to show that one of the steps of Algorithm 2.1 strictly decreases the energy. Let $\Sigma^0 = \Sigma$ and let Σ^1 and $\tilde{\Sigma}^1$ be the configurations obtained from Σ^0 after a single iteration of Algorithm 1.1 and Algorithm

2.1 respectively. Let $\varphi(x) = \mathbf{1}_{\Sigma^1} - \mathbf{1}_{\Sigma^0}$. By assumption, the support of $\varphi(x)$ has positive measure. Let $\tilde{\Sigma}^{1/2}$ be the intermediate set obtained after applying the first two steps of Algorithm 2.1 to Σ^0. Then

$$\mathbf{1}_{\tilde{\Sigma}^{1/2}} - \mathbf{1}_{\Sigma^0} = \max(\varphi(x), 0) = \varphi_+(x).$$

First suppose that the support of $\varphi_+(x)$ also has positive measure. Consider the change in energy

$$\mathrm{HC}_{\sqrt{\delta t}}(\tilde{\Sigma}^{1/2}, K_{\sqrt{\delta t}}) - \mathrm{HC}_{\sqrt{\delta t}}(\Sigma^0, K_{\sqrt{\delta t}}) \leq -\frac{1}{\sqrt{\delta t}} \int_D \varphi_+(x)(K_{\sqrt{\delta t}} * \varphi_+)(x)\, dx =$$

$$-\frac{1}{\sqrt{\delta t}} \int_D \varphi_+(x)(K^1_{\sqrt{\delta t}} * \varphi_+)(x)\, dx - \frac{1}{\sqrt{\delta t}} \sum_{\alpha \in L} (\widehat{\varphi}_+(\alpha))^2 \hat{K}^2(\alpha\sqrt{\delta t}).$$

It is enough to show that one of the two terms is strictly negative.

If $\hat{K}^2(0)$ is positive then we only need to show that $\hat{\varphi}_+(0) \neq 0$. This must be the case as φ_+ does not change signs and has support of positive measure.

If K^1 is positive in a neighborhood of the origin then there exists $\delta_0 > 0$ and $b_0 > 0$ such that $K^1(z) \geq b_0$ for all $z \in B(0, \delta_0)$. By the nonnegativity of K^1 we have

$$-\int_D \varphi_+(x)(K^1_{\sqrt{\delta t}} * \varphi_+)(x)\, dx \leq -b_0 \int_D \varphi_+(x) \int_{B(0,\delta_0)} \varphi_+(x + z\sqrt{\delta t})\, dz\, dx.$$

By the Lebesgue differentiation theorem

$$\lim_{\delta_0 \to 0} \frac{1}{m(B(0,\delta_0))} \int_{B(0,\delta_0)} \varphi_+(x + z\sqrt{\delta t})\, dz = 1$$

for almost every $x \in \mathrm{supp}(\varphi_+)$. Therefore

$$-b_0 \int_D \varphi_+(x) \int_{B(0,\delta_0)} \varphi_+(x + z\sqrt{\delta t})\, dz < 0.$$

On the other hand if $\varphi_+(x)$ has support of measure zero then $\tilde{\Sigma}^{1/2}$ is equal to Σ^0 except on a set of measure zero. Therefore

$$\frac{1}{(\delta t)^{\frac{d}{2}}} K\left(\frac{x}{\sqrt{\delta t}}\right) * \mathbf{1}_{\tilde{\Sigma}^{1/2}} = \frac{1}{(\delta t)^{\frac{d}{2}}} K\left(\frac{x}{\sqrt{\delta t}}\right) * \mathbf{1}_{\Sigma^0}.$$

It then follows that

$$\mathbf{1}_{\tilde{\Sigma}^1} - \mathbf{1}_{\tilde{\Sigma}^{1/2}} = \min(\varphi(x), 0) = \varphi_-(x).$$

Since φ_+ had support of measure zero, φ_- must have positive support. An analogous argument to the above implies that the energy must decrease.

□

2.2.2.1 Convergence of the single growth scheme with positive kernels

In addition to extending energy dissipation property to far more general kernels as described above in Proposition 2.2.1, Algorithm 2.1 maintains convergence to viscosity solution [31, 20] of the level set formulation of flow (1.8) in case the convolution kernel happens to be positive, with suitable decay and regularity, as we explain next. We will adapt to our new algorithm, Algorithm 2.1, the convergence argument that was given in [48] for the standard MBO scheme (1.1) for positive but otherwise fairly general convolution kernels. Hence, for the remainder of this subsection, we assume that kernel K satisfies the positivity, regularity, and decay properties (3.1) through (3.7) in [48], which are more stringent than assumptions needed elsewhere in this paper. In this framework, first threshold dynamics is extended from sets (binary functions) to $L^1(\mathbb{R}^d)$ in a level set–by–level set fashion: For $\varphi \in L^1(\mathbb{R})$, let

$$S_h\varphi(x) = (K_{\sqrt{h}} * \varphi)(x) \tag{2.6}$$

$$G_h\varphi(x) = \sup\{\lambda \in \mathbb{R} : S_h\mathbf{1}_{\{y:\varphi(y)>\lambda\}}(x) \geq 1/2\} \tag{2.7}$$

in keeping with the notation of [48]. If φ happens to be a characteristic function, applying G_h gives one step of the standard MBO algorithm (1.1) with time step size h. The new, single growth version of threshold dynamics described in Algorithm 2.1 can be written in terms of G_h as well. To that end, define the following two new operators:

$$G_h^{\uparrow}\varphi(x) = \max(\varphi(x), G_h\varphi(x)) \tag{2.8}$$

$$G_h^{\downarrow}\varphi(x) = \min(\varphi(x), G_h\varphi(x)). \tag{2.9}$$

Then, one step of Algorithm 2.1 applied to a function $\varphi(x)$ is given by

$$G_h^{\downarrow}G_h^{\uparrow}\varphi(x). \tag{2.10}$$

Next, define a piecewise constant–in–time approximation to the propagator of the limiting continuum flow:

$$Q_t^h = \left(G_h^{\downarrow}G_h^{\uparrow}\right)^{j-1} \text{ if } (j-1)h \leq t < jh \text{ with } j \in \mathbb{N}. \tag{2.11}$$

We can now state our convergence result.

Theorem 2.2.2. *Let $g : \mathbb{R}^d \to \mathbb{R}$ be a bounded, uniformly continuous function. Let $u : \mathbb{R}^d \times [0, \infty) \to \mathbb{R}$ be the unique viscosity solution of the PDE*

$$\begin{cases} u_t = -F(D^2u, Du) \\ u(x, 0) = g(x) \end{cases}$$

where F is given by

$$F(M, p) = -\left(\int_{p^\perp} K(x)\, dH^{d-1}(x)\right)^{-1} \left(\frac{1}{2}\int_{p^\perp} \langle Mx, x\rangle K(x)\, dH^{d-1}(x)\right) \tag{2.12}$$

for $d \times d$ symmetric matrices M and $p \in \mathbb{R}^d$. Then, for any $T \in [0, \infty)$,

$$Q_t^h g(x) \longrightarrow u(x, t) \text{ uniformly on } \mathbb{R}^d \times [0, T]$$

as $h \to 0^+$.

Operator $G_h^\downarrow G_h^\uparrow$ shares the following properties with G_h that are essential for the framework of [48]:

1. $G_h^\downarrow G_h^\uparrow(\rho \circ \varphi) = \rho \circ (G_h^\downarrow G_h^\uparrow \varphi)$ for all continuous, nondecreasing functions $\rho : \mathbb{R} \to \mathbb{R}$,
2. $G_h^\downarrow G_h^\uparrow \psi \geq G_h^\downarrow G_h^\uparrow \phi$ whenever $\psi \geq \phi$,
3. $G_h^\downarrow G_h^\uparrow(\phi + c) = G_h^\downarrow G_h^\uparrow \phi + c$, $G_h^\downarrow G_h^\uparrow c = c$, and $G_h^\downarrow G_h^\uparrow \phi(\cdot + y) = (G_h^\downarrow G_h^\uparrow \phi)(\cdot + y)$ for a constant $c \in \mathbb{R}$ and $y \in \mathbb{R}^d$.

Property 2, in particular, says that $G_h^\downarrow G_h^\uparrow$ is, just like G_h, monotone. Positivity of the convolution kernel is essential for the monotonicity property of the operator, hence our additional kernel restrictions in this subsection. Thanks to these properties, it follows from [5, 48] that to prove convergence of Algorithm 2.1, it is sufficient to establish the following consistency lemma:

Lemma 2.2.3. *For $\varphi \in C^2(D)$ for every $z \in D$ such that $D\varphi(z) \neq 0$ and for $\varepsilon > 0$ there exists $\delta > 0$ such that for all $x \in B(z, \delta)$ and $h \leq \delta$ we have the following inequalities:*

$$G_h^\downarrow G_h^\uparrow \varphi(x) \leq \varphi(x) + (\varepsilon - F(D^2\varphi(z), D\varphi(z)))h \tag{2.13}$$

$$G_h^\downarrow G_h^\uparrow \varphi(x) \geq \varphi(x) + (-\varepsilon - F(D^2\varphi(z), D\varphi(z)))h \tag{2.14}$$

Furthermore if $\varphi(x) = \sqrt{x^2+1}$ *then there exists* $\delta > 0$ *and* $C > 0$ *such that for every* x *and* $h \leq \delta$

$$G_h^{\downarrow} G_h^{\uparrow}(\varphi)(x) \leq \varphi(x) + Ch \tag{2.15}$$

$$G_h^{\downarrow} G_h^{\uparrow}(-\varphi)(x) \geq -\varphi(x) - Ch \tag{2.16}$$

Lemma 2.2.3 will follow from the following analogous statement for the operator G_h that can be found in [48], where it plays a pivotal role:

Lemma 2.2.4 ([48])**.** *If* $\varphi \in C^2(D)$ *then for every* $z \in D$ *such that* $D\varphi(z) \neq 0$ *and* $\varepsilon > 0$ *there exists* $\delta > 0$ *such that for all* $x \in B(z, \delta)$ *and* $h \leq \delta$ *we have the following inequalities:*

$$G_h\varphi(x) \leq \varphi(x) + (\varepsilon - F(D^2\varphi(z), D\varphi(z)))h \tag{2.17}$$

$$G_h\varphi(x) \geq \varphi(x) + (-\varepsilon - F(D^2\varphi(z), D\varphi(z)))h \tag{2.18}$$

Furthermore if $\varphi(x) = \sqrt{x^2+1}$ *then there exists* $\delta > 0$ *and* $C > 0$ *such that for every* x *and for* $h \leq \delta$

$$G_h(\varphi)(x) \leq \varphi(x) + Ch \tag{2.19}$$

$$G_h(-\varphi)(x) \geq -\varphi(x) - Ch \tag{2.20}$$

We now show how to obtain Lemma 2.2.3 from Lemma 2.2.4:

Proof of Lemma 2.2.3. First, observe that

$$G_h^{\downarrow} G_h^{\uparrow}\varphi \geq G_h\varphi \text{ for any } \varphi. \tag{2.21}$$

Indeed,

$$\begin{aligned} G_h^{\downarrow} G_h^{\uparrow}\varphi &= \min\Big(\max(G_h\varphi, \varphi)\,,\, G_h \max(G_h\varphi, \varphi)\Big) \\ &\geq \min\Big(\max(G_h\varphi, \varphi)\,,\, G_h\varphi\Big) \\ &\geq \min\Big(G_h\varphi\,,\, G_h\varphi\Big) = G_h\varphi \end{aligned}$$

where we used the monotonicity of G_h to get the first inequality. Inequality (2.14) now follows from (2.21) and inequality (2.18) of Lemma 2.2.4.

Next, observe that if all lower (or upper) level sets of φ are strictly convex, then $G_h^{\downarrow} G_h^{\uparrow} \varphi = G_h \varphi$. Thus, inequalities (2.15) & (2.16) follow immediately from inequalities (2.19) & (2.20) of Lemma 2.2.4.

What remains is inequality (2.13). Observe that if $\varepsilon - F(D^2\varphi(z), D\phi(z)) \geq 0$, then by the definition of $G_h^{\downarrow}$

$$G_h^{\downarrow} G_h^{\uparrow} \varphi \leq \max(\varphi, G_h \varphi) \tag{2.22}$$

$$\leq \max\left(\varphi(x), \varphi(x) + (\varepsilon - F(D^2\varphi(z), D\varphi(z)))h\right) \tag{2.23}$$

$$= \varphi(x) + (\varepsilon - F(D^2\varphi(z), D\varphi(z)))h. \tag{2.24}$$

Hence, all that remains is to establish inequality (2.13) in the case $\varepsilon - F(D^2\varphi(z), D\varphi(z)) < 0$. For the remainder of the argument we will write $F = F(D^2\varphi(z), D\varphi(z)))$ to simplify notation. By Lemma 2.2.4 there exists δ_0 such that for all $x \in \overline{B(z, \delta_0)}$ we have $G_h\varphi(x) \leq \varphi(x) + (\varepsilon/2 - F)h$. Then let

$$E_x = \{y : \varphi(y) \geq \varphi(x) + (\varepsilon - F)h\}$$

and $\theta(x) = S_h \mathbf{1}_{E_x}(x)$. It follows directly from (2.7) that $\theta(x) < 1/2$ for every $x \in \overline{B(z, \delta_0)}$. Thus $\theta_c = \sup_{x \in B(z, \delta_0)} \theta(x) < 1/2$. Therefore, recalling the definition (2.6) of S_h, we may choose δ so small that for every $x \in B(z, \delta)$ and $h \leq \delta$

$$\int_{B(z,\delta_0)^c} K_{\sqrt{h}}((x-y))\, dy < 1/2 - \theta_c.$$

Since $(\varepsilon - F)h < 0$ we know that for every $y \in B(z, \delta_0)$ we must have $G_h^{\uparrow}\varphi(y) = \varphi(y)$. Therefore if we consider the set

$$E_x^{\uparrow} = \{y : G_h^{\uparrow}\varphi(y) \geq \varphi(x) + (\varepsilon - F)h\}$$

it can only differ from E_x on $B(z, \delta_0)^c$. Thus $E_x^{\uparrow} \subset E_x \cup B(z, \delta_0)^c$. Taking $x \in B(z, \delta)$ we get the chain of inequalities

$$S_h \mathbf{1}_{E_x^{\uparrow}}(x) \leq S_h \mathbf{1}_{E_x}(x) + S_h \mathbf{1}_{B(z,\delta_0)^c}(x) \leq \theta_c + S_h \mathbf{1}_{B(z,\delta_0)^c}(x) < 1/2.$$

Therefore $G_h^{\downarrow} G_h^{\uparrow} \varphi(x) \leq \varphi(x) + (\varepsilon - F)h$ for $x \in B(z, \delta)$ and $h \leq \delta$ as desired.

□

2.2.3 Multiphase Single Growth Algorithm

In this section we describe versions of the single growth algorithm, inspired by the Gauss-Seidel scheme in [27], which dissipate the multiphase heat content energy, with quite general interfacial energies, at every step. Assume that D is partitioned into $N > 2$ sets, and recall that we denote the partition by $\mathbf{\Sigma} = (\Sigma_1, \dots \Sigma_N)$.

As noted in [25], the natural candidate for approximating the most general form of multiphase interfacial energy (1.7) in the style of the heat content Lyapunov functionals (1.18) and (1.21) is

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}, \mathbf{K}_{\sqrt{\delta t}}) = \frac{1}{\sqrt{\delta t}} \sum_{(i,j)\in\mathcal{I}_N} \int_{\Sigma_j} (K_{i,j})_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_i}\, dx \tag{2.25}$$

which requires choosing a possibly different convolution kernel for the anisotropy $\sigma_{i,j} : \mathbb{R}^N \to \mathbb{R}^+$ associated with each interface $(\partial\Sigma_i) \cap (\partial\Sigma_j)$ in the network. Here, we only require that each $K_{i,j}$ satisfies

$$K_{i,j}(x) = K_{j,i}(x) = K_{i,j}(-x) \tag{2.26}$$

for all $i \neq j$ and all x.

Following the general strategy described in [27] for deriving threshold dynamics-type algorithms from non-local approximate energies such as (2.25), we first extend energy (2.25) to functions $\mathbf{u} \in \mathcal{K}$, with time step δt:

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{u}, \mathbf{K}_{\sqrt{\delta t}}) = \frac{1}{\sqrt{\delta t}} \sum_{(i,j)\in\mathcal{I}_N} \int_{D} u_j (K_{i,j})_{\sqrt{\delta t}} * u_i\, dx \tag{2.27}$$

Then, a threshold dynamics algorithm can be systematically derived by linearizing (2.27) at a given configuration and minimizing it over the entire box constraint set $\mathcal{K}$. Fix a partition $\mathbf{\Sigma}$, and let $u_i = \mathbf{1}_{\Sigma_i}$. The linearization of relaxed energy (2.27) at $\mathbf{u} = (u_1, \dots, u_N)$, evaluated at some function $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_N)$ turns out to be:

$$L_{\mathbf{u},\sqrt{\delta t}}(\boldsymbol{\varphi}) = \mathrm{HC}_{\sqrt{\delta t}}(\mathbf{u}, \mathbf{K}_{\sqrt{\delta \mathbf{t}}}) + \frac{2}{\sqrt{\delta t}} \sum_{i=1}^{N} \int_{D} \varphi_i \sum_{j\neq i} (K_{i,j})_{\sqrt{\delta t}} * u_j\, dx \tag{2.28}$$

Dropping the factor $\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{u}, \mathbf{K}_{\sqrt{\delta \mathbf{t}}})$, which is constant in φ, we will write

$$\mathcal{L}_{\mathbf{u},\sqrt{\delta t}}(\boldsymbol{\varphi}) = \frac{2}{\sqrt{\delta t}} \sum_{i=1}^{N} \int_{D} \varphi_i \sum_{j\neq i} (K_{i,j})_{\sqrt{\delta t}} * u_j\, dx \tag{2.29}$$

Minimizing (2.29) over (1.23) yields the following algorithm from [25], which is the obvious extension of Algorithm 1.2 to normal dependent surface tensions:

Algorithm 2.2: ([25])

Given a time step size $\delta t > 0$, alternate the following steps:

1. Convolution:
$$\psi_i^k = \sum_{j \neq i} (K_{i,j})_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_j^k}. \tag{2.30}$$

2. Thresholding:
$$\Sigma_i^{k+1} = \left\{ x \,:\, \psi_i^k(x) \leq \min_{j \neq i} \psi_j^k(x) \right\}. \tag{2.31}$$

Algorithm 2.2 is natural, and appears to work well in practice; see [25] for some examples. However, the question of whether it in fact decreases the corresponding energy (2.25) for any choice of $\delta t > 0$ is now an even more complicated problem than in the case of Algorithm 1.2 for energy (1.21), not least because there are multiple ways to construct a convolution kernel corresponding to a given anisotropy: the stability of the algorithm is likely to depend not only on the properties of the surface tensions $\sigma_{i,j}$, but also the particular convolution kernels $K_{i,j}$ used to approximate them.

To make some headway, here we will instead consider new and slightly more expensive versions of Algorithm 2.2 that are motivated by the Gauss-Seidel version of Algorithm 1.2 given in [27] , as well as Algorithm 2.1 of the previous section. To that end, given a partition $\mathbf{\Sigma}$, define

$$i_{\min}(x) = \operatorname*{arg\,min}_{1 \leq i \leq N} \sum_{j \neq i} \int (K_{i,j})_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_j} \, dx$$

so that $x \in \Sigma_{i_{\min}(x)}$ after one step of Algorithm 2.2. Also, let $i_{\mathbf{\Sigma}}(x)$ denote the unique i such that $x \in \Sigma_i$, and let $\mathbf{e}_n \in \mathbb{R}^N$ denote the n^{th} standard basis vector. Then the direction of perturbation affected by Algorithm 2.2 on the current configuration is given by

$$\boldsymbol{\varphi}(x) = \mathbf{e}_{i_{\min}}(x) - \mathbf{e}_{i_\Sigma}(x) \tag{2.32}$$

for each $x \in D$. The simple linear structure of (2.29), makes it easy to see that (2.32) is the global minimizer of (2.29) among admissible perturbation directions $\boldsymbol{\varphi}$.

Below, we present Algorithms 2.3 and 2.4 which are the multiphase analogues of Algorithm 2.1. Algorithms 2.3 and 2.4 differ from Algorithm 2.2 by placing a single growth constraint on the perturbation direction φ. For each $x \in D$, if $i_{\min}(x)$ and $i_{\mathbf{\Sigma}}(x)$ fall into

certain classes (that depend on the iteration number) then $\boldsymbol{\varphi}(x)$ is chosen as in equation (2.32), otherwise $\boldsymbol{\varphi}(x)$ is set to $\mathbf{0}$. In other words, only a subset of the points $x \in D$ are redistributed among the phases as indicated by (2.32). Although we must pay the cost of slightly more convolutions per time step, the essential advantage of this approach is that each update moves the configuration in a more reliable descent direction. It turns out that this simple modification guarantees energy dissipation for a much wider class of kernels as we describe below.

Algorithm 2.3: Multi-phase super single growth

Given an initial partition of D into N sets $\boldsymbol{\Sigma}^0 = \{\Sigma_i^0\}_{i=1}^N$ and a time step δt the $(k+1)^{th}$ iteration $\boldsymbol{\Sigma}^{k+1}$ is obtained from $\boldsymbol{\Sigma}^k$ by a series of substeps indexed by $(m,n) \in \mathcal{I}_N$. For $(m,n) \neq (1,2)$ let $p(m,n)$ denote the predecessor of (m,n) in the dictionary ordering of $\mathcal{I}_N$ and define $\boldsymbol{\Sigma}^{k,p(1,2)} := \boldsymbol{\Sigma}^k$ and $\boldsymbol{\Sigma}^{k+1} := \boldsymbol{\Sigma}^{k,(N,N-1)}$. Then $\boldsymbol{\Sigma}^{k,(m,n)}$ is obtained from $\boldsymbol{\Sigma}^{k,p(m,n)}$ as follows:

1. For each $(i,j) \in \mathcal{I}_N$ form the convolutions:

$$\psi_i^{k,(m,n)}(x) = \sum_{j \neq i} (K_{i,j})_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_j^{k,p(m,n)}} \tag{2.33}$$

2. Threshold the m^{th} function:

$$G^{k,(m,n)} = \{x \in D : \min_i \psi_i^{k,(m,n)}(x) = \psi_m^{k,(m,n)}(x)\} \tag{2.34}$$

3. Grow set m into set n only:

$$\Sigma_m^{k,(m,n)} = \Sigma_m^{k,p(m,n)} \cup (G^{k,(m,n)} \cap \Sigma_n^{k,p(m,n)}) \tag{2.35}$$

4. Update set n:

$$\Sigma_n^{k,(m,n)} = \Sigma_n^{k,p(m,n)} \setminus (G^{k,(m,n)} \cap \Sigma_n^{k,p(m,n)}) \tag{2.36}$$

Proposition 2.2.5. *Suppose that each kernel $K_{i,j}$ may be split into a sum $K_{i,j} = K_{i,j}^1 + K_{i,j}^2$ such that $K_{i,j}^1 \geq 0$ almost everywhere and $\widehat{K}_{i,j}^2 \geq 0$ almost everywhere. In that case, Algorithm 2.3 dissipates the multi-phase heat content energy (2.25) at each substep. Furthermore if for every $(i,j) \in \mathcal{I}_N$ either $K_{i,j}^1$ is positive in a neighborhood of the origin or $\hat{K}_{i,j}^2(0)$ is positive then if an iteration of Algorithm 2.2 changes a configuration $\boldsymbol{\Sigma}$ by a set of positive measure, an iteration of Algorithm 2.3 also changes $\boldsymbol{\Sigma}$ by a set of positive measure and strictly decreases the energy.*

Proof. We show that at each substep the energy is dissipated moving from $\boldsymbol{\Sigma}^{k,p(m,n)}$ to $\boldsymbol{\Sigma}^{k,(m,n)}$. Set $\boldsymbol{\varphi}^{k,(mn)} = \mathbf{1}_{\boldsymbol{\Sigma}^{k,(m,n)}} - \mathbf{1}_{\boldsymbol{\Sigma}^{k,p(m,n)}}$. Using the quadratic structure of the energy

we may write

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}^{k,(m,n)}, \mathbf{K}_{\sqrt{\delta t}}) - \mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}^{k,p(m,n)}, \mathbf{K}_{\sqrt{\delta t}}) =$$

$$\mathcal{L}_{\mathbf{\Sigma}^{k,p(m,n)},\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,(m,n)}) + \mathrm{HC}_{\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,(m,n)}, \boldsymbol{K}_{\sqrt{\delta t}}). \tag{2.37}$$

where $\mathcal{L}_{\mathbf{\Sigma}^{k,m},\sqrt{\delta t}}$ is given in equation (2.29). Our perturbation $\boldsymbol{\varphi}^{k,(m,n)}$ is chosen to be the global minimum of $\mathcal{L}_{\mathbf{\Sigma}^{k,m},\sqrt{\delta t}}$ over all perturbations that change phases m and n only. Clearly $\mathcal{L}_{\mathbf{\Sigma}^{k,m},\sqrt{\delta t}}(\mathbf{0}) = 0$, thus it follows that $\mathcal{L}_{\mathbf{\Sigma}^{k,p(m,n)},\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,(m,n)}) \leq 0$. Therefore it suffices to show that $\mathrm{HC}_{\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,(m,n)}, K_{\sqrt{\delta t}}) \leq 0$. This term is given by

$$\mathrm{HC}_{\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,(m,n)}) = \frac{1}{\sqrt{\delta t}} \sum_{(i,j)\in\mathcal{I}_N} \int_D \varphi_i^{k,(m,n)}(x) \Big((K_{i,j})_{\sqrt{\delta t}} * \varphi_j^{k,(m,n)} \Big)(x)\, dx. \tag{2.38}$$

This formula actually simplifies dramatically as $\varphi_i^{k,(m,n)} \equiv 0$ unless $i = m$ or $i = n$. Furthermore $\varphi_m^{k,(m,n)}$ is nonnegative pointwise and $\varphi_n^{k,(m,n)} = -\varphi_m^{k,(m,n)}$. Thus nearly every term of $\mathrm{HC}_{\sqrt{\delta t}}(\varphi^{k,(m,n)})$ is zero and we get

$$\mathrm{HC}_{\sqrt{\delta t}}(\varphi^{k,(m,n)}) = -2 \int_D \varphi_m^{k,(m,n)}(x)((K_{m,n})_{\delta t} * \varphi_m^{k,(m,n)})(x)\, dx \tag{2.39}$$

Recalling equation (2.5) and the subsequent argument in Proposition 2.2.1 energy dissipation is proven.

Now we turn to the second statement. Let $\mathbf{\Sigma}^0 = \mathbf{\Sigma}$. Let $\mathbf{\Sigma}^1$ be the configuration obtained from a single iteration of Algorithm 2.2 and let $\mathbf{\Sigma}^{0,(m,n)}$ be the configurations obtained from the substeps of Algorithm 2.3. As before set

$$\boldsymbol{\varphi}(x) = \mathbf{1}_{\mathbf{\Sigma}^1}(x) - \mathbf{1}_{\mathbf{\Sigma}^0}(x)$$

and

$$\boldsymbol{\varphi}^{0,(m,n)}(x) = \mathbf{1}_{\mathbf{\Sigma}^{0,(m,n)}}(x) - \mathbf{1}_{\mathbf{\Sigma}^{0,p(m,n)}}(x).$$

We need to show that for some $(m,n) \in \mathcal{I}_N$ the function $\varphi_m^{0,(m,n)}(x)$ has support of positive measure. Note that if $\varphi_m^{0,(m,n)}(x)$ has support of positive measure then (2.39) will be strictly negative, implying that the energy strictly decreases.

Suppose that for every $(m,n) \in \mathcal{I}_N$ the function $\varphi_m^{0,(m,n)}$ has support of zero measure. In this case it follows that no set has grown or shrunk by more than a set of measure zero. Thus for any function $f \in L^1(D)$ and any label $1 \leq i \leq N$ we have

$$\int_{\Sigma_i^0} f(x)\, dx = \int_{\Sigma_i^{0,(1,2)}} f(x)\, dx = \cdots = \int_{\Sigma_i^{0,(N-1,N)}} f(x)\, dx$$

As a result we may compute every convolution in the substeps of Algorithm 2.3 against $\mathbf{\Sigma}^0$ without changing the result. This allows us to write $\varphi_m^{0,(m,n)}$ in terms of $\boldsymbol{\varphi}$:

$$\varphi_m^{0,(m,n)} = \max(\varphi_m(x), 0)|\min(\varphi_n(x), 0)|.$$

If we then sum over $n \neq m$ we get

$$\sum_{n \neq m} \varphi_m^{0,(m,n)} = \max(\varphi_m(x), 0).$$

The support of $\boldsymbol{\varphi}$ may be decomposed as

$$\operatorname{supp}(\boldsymbol{\varphi}) = \bigcup_{1 \leq m \leq N} \operatorname{supp}\big(\max(\varphi_m(x), 0)\big).$$

Thus $\boldsymbol{\varphi}$ has support of measure zero a contradiction.

□

Next we describe a variant of Algorithm 2.3 that requires fewer convolutions but imposes a more restrictive condition on the kernels.

Proposition 2.2.6. *Suppose that each kernel $K_{i,j}$ may be split into a sum $K_{i,j} = K_{i,j}^1 + K_{i,j}^2$ such that $K_{i,j}^1 \geq 0$ almost everywhere, $\widehat{K}_{i,j}^2 \geq 0$ and for almost every $x \in \mathbb{R}^d$ and i, j, k pairwise different we have the pointwise triangle inequality*

$$K_{i,k}(x) \leq K_{i,j}(x) + K_{j,k}(x) \tag{2.44}$$

then Algorithm 2.4 dissipates the energy (2.25) at each step. Furthermore if for every $(i, j) \in \mathcal{I}_N$ either $K_{i,j}^1$ is positive in a neighborhood of the origin or $\hat{K}_{i,j}^2(0)$ is positive then if an iteration of Algorithm 2.2 changes a configuration $\mathbf{\Sigma}$ by a set of positive measure, an iteration of Algorithm 2.4 also changes $\mathbf{\Sigma}$ by a set of positive measure and strictly decreases the energy.

Proof. We proceed by showing that the energy is dissipated moving from substep $\mathbf{\Sigma}^{k,m}$ to $\mathbf{\Sigma}^{k,m+1}$. Let $\boldsymbol{\varphi}^{k,m+1}(x) = \mathbf{1}_{\mathbf{\Sigma}^{k,m+1}}(x) - \mathbf{1}_{\mathbf{\Sigma}^{k,m}}(x)$.

As in the argument of Proposition 2.2.5 the change in energy

$$\mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}^{k,m+1}, \mathbf{K}_{\sqrt{\delta t}}) - \mathrm{HC}_{\sqrt{\delta t}}(\mathbf{\Sigma}^{k,m}, \mathbf{K}_{\sqrt{\delta t}})$$

will be nonnegative as long as the quadratic term in the difference $\mathrm{HC}_{\sqrt{\delta t}}(\boldsymbol{\varphi}^{k,m}, K_{\delta t})$ is nonnegative. Equations (2.42) and (2.43) show that $\varphi_{m+1}^{k,m+1}(x)$ is pointwise nonnegative and

Algorithm 2.4: Multi-phase single growth

Given an initial partition of D into N sets $\mathbf{\Sigma}^0 = \{\Sigma_i^0\}_{i=1}^N$ and a time step δt the $(k+1)^{th}$ iteration $\mathbf{\Sigma}^{k+1}$ is obtained from $\mathbf{\Sigma}^k$ by computing the following N substeps $\mathbf{\Sigma}^{k,0}, \ldots, \mathbf{\Sigma}^{k,N}$ where $\mathbf{\Sigma}^{k,0} := \mathbf{\Sigma}^k$ and $\mathbf{\Sigma}^{k+1} := \mathbf{\Sigma}^{k,N}$.
For each $0 \leq m \leq \ell - 1$ the partitions $\mathbf{\Sigma}^{k,m+1}$ are obtained from $\mathbf{\Sigma}^{k,m}$ as follows:

1. For each $(i,j) \in \mathcal{I}_N$ form the convolutions:

$$\psi_i^{k,m+1}(x) = \sum_{j \neq i} (K_{i,j})_{\sqrt{\delta t}} * \mathbf{1}_{\Sigma_j^{k,m}} \tag{2.40}$$

2. Threshold the $(m+1)^{th}$ function:

$$G^{k,m+1} = \{x \in D : \min_i \psi_i^{k,m+1}(x) = \psi_{m+1}^{k,m+1}(x)\} \tag{2.41}$$

3. Grow the $(m+1)^{th}$ set:

$$\Sigma_{m+1}^{k,m+1} = \Sigma_{m+1}^{k,m} \cup G^{k,m+1} \tag{2.42}$$

4. Update the other sets:

$$\Sigma_i^{k,m+1} = \Sigma_i^{k,m} \setminus G^{k,m+1} \quad \forall i \neq m+1 \tag{2.43}$$

$\varphi_i^{k,m+1}(x)$ is pointwise nonpositive for $i \neq m+1$. In fact $\varphi_{m+1}^{k,m+1}(x) = -\sum_{i\neq m+1} \varphi_i^{k,m+1}(x)$. Plugging this into the formula for HC from equation (2.38) and defining

$$\mathcal{I}_N(m+1) = \{(i,j) \in \mathcal{I}_N : i,j \neq m+1\}$$

we get

$$-\sum_{i\neq m+1}\sum_{j\neq m+1}\int_D |\varphi_i^{k,m+1}(x)|\left(\left((K_{i,m+1})_{\sqrt{\delta t}} + (K_{j,m+1})_{\sqrt{\delta t}}\right) * |\varphi_j^{k,m+1}|\right)(x)\,dx+$$

$$\sum_{(i,j)\in\mathcal{I}_N(m+1)}\int_D |\varphi_i^{k,m+1}(x)|\left((K_{i,j})_{\sqrt{\delta t}} * |\varphi_j^{k,m+1}|\right)(x)\,dx.$$

Applying (2.44) the above is

$$\leq -2\sum_{i\neq m+1}\int_D |\varphi_i^{k,m+1}(x)|\left((K_{i,m+1})_{\sqrt{\delta t}} * |\varphi_i^{k,m+1}|\right)(x)\,dx$$

The remainder of the argument proceeds exactly as in the proof of Proposition 2.2.5

□

2.3 Convergence of non-local energies

In [27], Gamma convergence of the Lyapunov functional (1.21) to the interfacial energy (1.20) is established for radially monotonic and symmetric, nonnegative kernels. However, Algorithms 2.1, 2.3, and 2.4 guarantee energy dissipation for a much larger class of kernels. Thus it is desirable to have a more general Gamma convergence result.

In this section, we establish the Gamma limit of the more general heat content energy (2.27) for a much wider class of kernels, including *sign changing* kernels. The key property that we require of the kernel is a strong *positive core* near the origin; otherwise, the kernel is free to oscillate above and below zero at the outskirts. The positive core ensures that any negative mass further out will be sufficiently counterbalanced. To that end, recalling the definition of $\mathcal{K}$ let

$$BV_{\mathcal{B}} = \left\{\mathbf{u} \in \mathcal{K} : u_i(x) \in \{0,1\} \text{ and } u_i \in BV(D) \text{ for } i \in \{1,2,\ldots,N\}\right\}$$

and for any $\mathbf{u} \in \mathcal{K}$ define the energy

$$E(\mathbf{u}, \boldsymbol{\sigma}) = \begin{cases} \displaystyle\sum_{(i,j)\in\mathcal{I}_N} \int_D \sigma_{i,j}(\nabla u_i) + \sigma_{i,j}(\nabla u_j) - \sigma_{i,j}\big(\nabla(u_i + u_j)\big) & \text{if } \mathbf{u} \in BV_{\mathcal{B}}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.45}$$

Theorem 2.3.1. *Suppose that each kernel K_{ij} can be written as a positive linear combination of m kernels $K_{ij}(z) = \sum_{r=1}^m \sigma_{i,j}^r K^r(z)$ where each K^r satisfies (1.16) and the constants σ_{ij}^r satisfy the triangle inequality (1.13) for each r. In addition, assume that there exist positive constants a_r, α_r, β_r such that the following conditions hold:*

1. $\alpha(\frac{2}{a_r})^{d+2} \leq \beta_r$,
2. $K^r(z) \geq \beta_r$ *for* $|z| \leq a_r$,
3. $|\min(K^r(z), 0)| \leq \alpha_r |z|^{-(d+2)}$ *for all* z.

If we define

$$\sigma_{i,j}(n) = \sum_{r=1}^m \sigma_{ij}^r \int_{\mathbb{R}^d} K^r(z)|z \cdot n|\, dz$$

then as $\varepsilon \to 0$ the Lyapunov functionals $HC_\varepsilon(\cdot, \boldsymbol{K}_\varepsilon)$ given in (2.27) Gamma converge in the L^1 topology over $\mathcal{K}$ to the energy $E(\cdot, \boldsymbol{\sigma})$ given in (2.45). Furthermore if for some sequence $\mathbf{u}_\varepsilon$ we have $\sup_{\varepsilon>0} HC_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon) < \infty$ then $\mathbf{u}_\varepsilon$ is pre-compact in $L^1(D)$ and the set of accumulation points is contained in $BV_{\mathcal{B}}(D)$.

Before we give the proof of Theorem 2.3.1 a number of remarks are in order. Although our result seems very general, the networks of surface tensions $\boldsymbol{\sigma}$ that have the form $\sigma_{ij}(n) = \sum_{r=1}^m c_{ij}^r \int_{\mathbb{R}^d} K^r(z)|z \cdot n| dz$ appear to be somewhat limited. In particular, they seem to be less general than some of the known classes of surface tension networks $\boldsymbol{\sigma}$ that guarantee lower semi-continuity of energy (2.45). In principle, one should hope that a more general result is possible – lower semi-continuity is the only *obvious* necessary condition for (2.45) to arise as the Gamma limit of a sequence of functionals. Worse, as it turns out, the admissible sign-changing kernels K^r can only produce zonoidal anisotropies (we will see why below). Nonetheless, our result is the most general to date, and we hope our arguments will spur further developments on this front.

The proof of Theorem 2.3.1 will be built over the following lemmas and propositions. First, we will prove the theorem in the case that $m = 1$ (i.e. just a single kernel). To start, we will restrict ourselves to kernels that satisfy (1.16) and are positive in a neighborhood

of the origin. Then we will show that our energies satisfy an inequality that will allow us to swap out sign-changing kernels with non-negative kernels. Finally, we will show that the general case $m > 1$ (i.e. surface tensions built from a linear combination of kernels) follows immediately from the case $m = 1$. Since we will be primarily working in the case $m = 1$, we will suppress the subscript and superscript r's in what follows.

Lemma 2.3.2. *Suppose that K is a nonnegative kernel that satisfies (1.16) and is positive in a neighborhood of the origin. If for some sequence $\mathbf{u}_\varepsilon$ we have $\sup_{\varepsilon>0} \boldsymbol{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon) < \infty$ then $\mathbf{u}_\varepsilon$ is pre-compact in $L^1(D)$ and the set of accumulation points is contained in $BV_{\mathcal{B}}(D)$.*

Proof. K is strictly positive in a neighborhood of the origin thus there exists some $s, t \in (0, 1)$ such that $K(z) \geq s$ for all $|z| \leq t$. Let $J(z) = cs(1 - |z/t|)$ for $|z| < t$ and 0 otherwise, where c is chosen so that J has unit mass. Then $\frac{1}{c}J(z) \leq K(z)$ for all $z \in \mathbb{R}^d$. Therefore

$$\sup_{\varepsilon>0} \mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \frac{1}{c}\boldsymbol{J}_\varepsilon) \leq \sup_{\varepsilon>0} \mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon).$$

Since the energy is linear in the kernel it follows that $\sup_{\varepsilon>0} \mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{J}_\varepsilon)$ is bounded. In addition $|\nabla J(z)| = \frac{cs}{t}$ for $|z| < t$ and 0 for $|z| > t$. Thus $|\nabla J(z)| \leq \frac{2}{t}J(z/2)$ for all z. Now J fits into the framework of Lemma 5 in [27], which gives the desired result.

□

Next we show that the heat content converges pointwise to $E(\mathbf{u}, \sigma)$. By choosing $\boldsymbol{u}_\varepsilon = \boldsymbol{u}$ as the recovery sequence, the pointwise convergence will immediately give us the $\limsup$ inequality. If $\mathbf{u} \notin BV_{\mathcal{B}}(D)$ then the pointwise convergence follows from Lemma 2.3.2. Indeed we must have $\lim_{\varepsilon\to 0} \mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon) = \infty$ for otherwise the constant sequence $\mathbf{u}$ would have an accumulation point in $BV_{\mathcal{B}}(D)$ implying $\mathbf{u} \in BV_{\mathcal{B}}(D)$.

For $\mathbf{u} \in BV_{\mathcal{B}}(D)$ we recall Lemma 4 from [27] which gives pointwise convergence under very mild conditions on the kernel. Although the argument in [27] is given for radially symmetric kernels, the modification to general kernels is straight forward.

Lemma 2.3.3. *(Lemma 4 in [27]) Let K be a kernel satisfying (1.16) and $\mathbf{u} \in BV_{\mathcal{B}}(D)$ then $\lim_{\varepsilon\to 0} \boldsymbol{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon) = E(\mathbf{u}, \boldsymbol{\sigma})$.*

To complete the Gamma convergence argument for nonnegative kernels we only have left to prove the $\liminf$ inequality. A key tool that we will need is Lemma 3 from [27] that says that integer scalings of the parameter ε are guaranteed to decrease the energy.

Lemma 2.3.4. *If the kernel K is nonnegative and the surface tensions $\boldsymbol{\sigma}$ satisfy the triangle inequality (1.13) then $\boldsymbol{HC}_{N\varepsilon}(\boldsymbol{u}, \boldsymbol{K}_{N\varepsilon}) \leq \boldsymbol{HC}_\varepsilon(\boldsymbol{u}, \boldsymbol{K}_\varepsilon)$.*

Now we are ready to present the lim inf argument.

Proposition 2.3.5. *If K satisfies (1.16) and in addition K is nonnegative then for any sequence $\mathbf{u}_\varepsilon$ converging to $\mathbf{u}$ in L^1 the inequality $\liminf_{\varepsilon\to 0} \boldsymbol{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon) \geq E(\mathbf{u}, \boldsymbol{\sigma})$ holds.*

Proof. K is pointwise nonnegative therefore if we fix some $L > 0$ and let

$$K^L(z) = K(z)\mathbf{1}_{B(0,L)}(z)$$

then we decrease the energy by replacing $\mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, K_\varepsilon)$ with $\mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}^L_\varepsilon)$. Now fix $\delta > 0$ and for each ε let $\delta_\varepsilon = n_\varepsilon \varepsilon$ where $n_\varepsilon \in \mathbb{Z}_+$ is chosen such that $|\delta - \delta_\varepsilon|$ is minimized. It follows immediately that $|\delta - \delta_\varepsilon| \leq \varepsilon/2$. Since δ_ε is obtained from ε by an integer scaling we may use Lemma 2.3.4 to get the inequality

$$\mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}^L_\varepsilon) \geq \mathrm{HC}_{\delta_\varepsilon}(\mathbf{u}_\varepsilon, \boldsymbol{K}^L_{\delta_\varepsilon}).$$

Now we wish to replace δ_ε with δ. Thus we must estimate the resulting error

$$R_{\varepsilon,\delta} = |\mathrm{HC}_{\delta_\varepsilon}(\mathbf{u}_\varepsilon, \boldsymbol{K}^L_{\delta_\varepsilon}) - \mathrm{HC}_\delta(\mathbf{u}_\varepsilon, \boldsymbol{K}^L_\delta)|.$$

We have:

$$\begin{aligned} R_{\varepsilon,\delta} &\leq \sum_{(i,j)\in\mathcal{I}_N} \sigma_{i,j} \int_D \int_{B(0,L)} u_{\varepsilon,i}(x) u_{\varepsilon,j}(x+z) \left| \delta_\varepsilon^{-(d+1)} K\left(\frac{z}{\delta_\varepsilon}\right) - \delta^{-(d+1)} K\left(\frac{z}{\delta}\right) \right| dz\, dx \\ &\leq sN^2 \int_D \int_{B(0,L/\delta)} \delta^{-1} \left| \left(\frac{\delta}{\delta_\varepsilon}\right)^{(d+1)} K\left(\frac{\delta}{\delta_\varepsilon} z\right) - K(z) \right| dz\, dx \end{aligned}$$

where $s = \max_{(i,j)\in\mathcal{I}_N} \sigma_{ij}$. Smooth functions are dense in $L^1(\mathbb{R}^d)$, thus for any small $\gamma > 0$ we may find a smooth f independent of ε such that

$$sN^2 \int_D \int_{B(0,L/\delta)} \delta^{-1} \left(\left(\frac{\delta}{\delta_\varepsilon}\right)^{(d+1)} \left| K\left(\frac{\delta}{\delta_\varepsilon} z\right) - f\left(\frac{\delta}{\delta_\varepsilon} z\right) \right| + |f(z) - K(z)| \right) dz\, dx < \gamma.$$

The spaces D and $B(0, L/\delta)$ have finite measure thus uniform continuity shows that

$$\lim_{\varepsilon\to 0} \int_D \int_{B(0,L/\delta)} \delta^{-1} \left| \left(\frac{\delta}{\delta_\varepsilon}\right)^{(d+1)} f\left(\frac{\delta}{\delta_\varepsilon} z\right) - f(z) \right| dz\, dx = 0.$$

It then follows that

$$\lim_{\varepsilon\to 0} R_{\varepsilon,\delta} = 0.$$

Combining the above with the L^1 convergence of $\mathbf{u}_\varepsilon$ to $\mathbf{u}$ we obtain:

$$\liminf_{\varepsilon\to 0} \mathrm{HC}_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon^L) \geq \liminf_{\varepsilon\to 0} \mathrm{HC}_\delta(\mathbf{u}_\varepsilon, \boldsymbol{K}_\delta^L) = \mathrm{HC}_\delta(\mathbf{u}, \boldsymbol{K}_\delta^L)$$

Now if we allow δ to go to zero the question of the liminf inequality has been reduced to the question of pointwise convergence of the functional. However this is already covered above thus

$$\lim_{\delta\to 0} \mathrm{HC}_\delta(\mathbf{u}, \boldsymbol{K}_\delta^L) = E(\mathbf{u}, \boldsymbol{\sigma}^L)$$

where

$$\sigma_{i,j}^L(n) = \sigma_{i,j} \int_{\mathbb{R}^d} K^L(z)|z\cdot n|\, dz.$$

By monotone convergence $\lim_{L\to\infty} E(\mathbf{u}, \boldsymbol{\sigma}^L) = E(\mathbf{u}, \boldsymbol{\sigma})$.

□

This completes the proof of Theorem 2.3.1 for nonnegative kernels that are positive in a neighborhood of the origin. To extend the result to kernels that change sign we show that it is possible to rearrange the mass of the kernel so that it becomes nonnegative, while also decreasing the energy functional (2.27). It is essential that this process does not change the limiting energy $E(\mathbf{u}, \boldsymbol{\sigma})$ and that the rearranged kernel still is positive in a neighborhood of the origin and satisfies (1.16). The following lemma shows that these goals can be accomplished simultaneously. Note that since the limiting anisotropy also corresponds to a nonnegative kernel, the anisotropy must be zonoidal.

Lemma 2.3.6. *Suppose that K satisfies the conditions in Theorem 2.3.1. Then there exists a nonnegative kernel $\tilde{K}$, which is positive in a neighborhood of the origin, satisfies (1.16), satisfies the inequality $HC_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon) \geq HC_\varepsilon(\mathbf{u}, \tilde{\boldsymbol{K}}_\varepsilon)$, and for every $n \in S^{d-1}$*

$$\int_{\mathbb{R}^d} |z\cdot n| K(z) dz = \int_{\mathbb{R}^d} |z\cdot n| \tilde{K}(z)\, dz.$$

Proof. Split K into $K^+ = \max(K, 0)$ and $K^- = \min(K, 0)$. Recall the constants a, α, β from Theorem 2.3.1. For $j \in \mathbb{Z}_+$ let

$$A_j = \{z \in \mathbb{R}^d : |z| \in (a2^{j-1}, a2^j)\}$$

Define

$$\tilde{K}(z) = K^+(z) + \sum_{j=1}^{\infty} 2^{j(d+1)} K^-(2^j z)\mathbf{1}_{A_j}(2^j z).$$

From this construction we see that $\tilde{K}$ is possibly negative only for z satisfying $|z| \in$

$(a/2, a)$. Recall that $|K^-(z)| \leq \alpha|z|^{-(d+2)}$ and $K(z) \geq \beta$ for $|z| < a$. Choose some z_0 such that $|z_0| \in (a/2, a)$ then using these inequalities we see

$$\tilde{K}(z_0) = K^+(z_0) + \sum_{j=1}^{\infty} 2^{j(d+1)} K^-(2^j z_0) \mathbf{1}_{A_j}(2^j z_0) \geq \beta - \alpha(\frac{2}{a})^{d+2} \sum_{j=1}^{\infty} 2^{-j} = \beta - \alpha(\frac{2}{a})^{d+2} \geq 0.$$

It is clear that $\tilde{K}$ satisfies the symmetry condition $\tilde{K}(z) = \tilde{K}(-z)$ since K^+ and K^- satisfy this condition and the A_j are radially symmetric sets. Near the origin $\tilde{K} = K^+$ so there must be a neighborhood where $\tilde{K}$ is strictly positive.

Next we recall that $\mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon)$ is linear in the kernel. Thus, using Fubini's theorem we may write

$$\mathrm{HC}_\varepsilon\big(\mathbf{u}, \tilde{\boldsymbol{K}}_\varepsilon(z)\big) = \mathrm{HC}_\varepsilon\big(\mathbf{u}, \boldsymbol{K}_\varepsilon^+(z)\big) + \sum_{j=1}^{\infty} 2^{j(d+1)} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_\varepsilon(2^j z)\big).$$

All of the terms in the infinite sum are negative, thus if we decrease their magnitude the overall energy will increase. By Lemma 2.3.4 we know that

$$|\mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_\varepsilon(2^j z)\big)| \geq |\mathrm{HC}_{\varepsilon 2^j}\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_{\varepsilon 2^j}(2^j z)\big)|.$$

Writing out the formula for the energy functional

$$\begin{aligned}
&\mathrm{HC}_{\varepsilon 2^j}\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_{\varepsilon 2^j}(2^j z)\big) \\
&\quad = \sum_{(i,j) \in \mathcal{I}_N} \sigma_{i,j} \frac{1}{\varepsilon 2^j} \int_D u_i(x) \int_{\mathbb{R}^d} \frac{1}{(\varepsilon 2^j)^d} K(\frac{z}{\varepsilon}) \mathbf{1}_{A_j}(z/\varepsilon) u_j(x+z)\, dz\, dx \\
&\quad = \frac{1}{2^{j(d+1)}} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_\varepsilon(z)\big)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathrm{HC}_\varepsilon\big(\mathbf{u}, \boldsymbol{K}_\varepsilon^+(z)\big) + \sum_{j=1}^{\infty} 2^{j(d+1)} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_\varepsilon(2^j z)\big) \\
\leq \mathrm{HC}_\varepsilon\big(\mathbf{u}, \boldsymbol{K}_\varepsilon^+(z)\big) + \sum_{j=1}^{\infty} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^- \mathbf{1}_{A_j})_\varepsilon(z)\big).
\end{aligned}$$

Now note that $K_\varepsilon^+(z) + \sum_{j=1}^\infty (K^-\mathbf{1}_{A_j})_\varepsilon(z) = K_\varepsilon(z)$ almost everywhere. Therefore

$$\mathrm{HC}_\varepsilon\big(\mathbf{u}, \boldsymbol{K}_\varepsilon^+(z)\big) + \sum_{j=1}^\infty \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^-\mathbf{1}_{A_j})_\varepsilon(z)\big) = \mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon)$$

and thus we have the desired result

$$\mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon) \geq \mathrm{HC}_\varepsilon(\mathbf{u}, \tilde{\boldsymbol{K}}_\varepsilon).$$

It remains to show $\int_{\mathbb{R}^d} |z\cdot n|K(z)dz = \int_{\mathbb{R}^d} |z\cdot n|\tilde{K}(z)\,dz$. This reduces to showing that

$$\int_{\mathbb{R}^d} |z\cdot n|K^-(z)\,dz = \sum_{j=1}^\infty 2^{j(d+1)} \int_{\mathbb{R}^d} |z\cdot n|K^-(z2^j)\mathbf{1}_{A_j}(z2^j)\,dz.$$

Changing variables $z' = z2^j$ for each integral on the right hand side and then summing the results gives the equality. The equality implies that $z\tilde{K} \in L^1(\mathbb{R}^d)$ and since $\tilde{K} = K^+$ near the origin it also follows that $\tilde{K} \in L^1(\mathbb{R}^d)$.

□

This completes the proof of Theorem 2.3.1 in the case that $m = 1$. To obtain the general case $m > 1$ we note that we may write $\mathrm{HC}_\varepsilon(\boldsymbol{u}, \boldsymbol{K}_\varepsilon) = \sum_{r=1}^m \mathrm{HC}_\varepsilon(\boldsymbol{u}, \boldsymbol{\sigma}^r K_\varepsilon^r)$. Our above Gamma convergence proof applies to each term inside the sum individually. Since we use the same recovery sequence $\boldsymbol{u}_\varepsilon = \boldsymbol{u}$ for each term, we may conclude that the entire sum Gamma converges to (2.45) completing the argument. Thus, Theorem 2.3.1 is now proven.

Note that the kernel inequalities given in Theorem 2.3.1 were only used to show that a certain rearrangement of the negative mass of the kernel K could produce a nonnegative kernel $\tilde{K}$. Indeed the actual necessary conditions on K needed to find a properly rearranged nonnegative $\tilde{K}$ are much weaker than the given inequalities. However, a necessary and sufficient condition is extremely difficult to describe in terms of the physical properties of the kernel. Thus in the next lemma we instead describe all rearrangements that will decrease (1.22) and preserve $E(\mathbf{u}, \sigma)$ along with (1.16) and positivity in a neighborhood of the origin. As a result if for some kernel K one of the following rearrangements produces a nonnegative $\tilde{K}$ then $\mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon)$ Gamma converges to $E(\mathbf{u}, \boldsymbol{\sigma})$.

Lemma 2.3.7. *Given a kernel K that satisfies (1.16) and has a neighborhood of the origin U such that $U = -U$ and $K(U) \subset (0, \infty)$, let $\{\Omega_m^+\}$ and $\{\Omega_j^-\}$ be measurable decomposition of $\mathrm{supp}(K^+) \setminus U$ and $\mathrm{supp}(K^-)$ respectively and let $\{\psi_m^+\}$ and $\{\psi_j^-\}$ be sequences of nonnegative functions supported on $\mathrm{supp}(K^+)$ and $\mathrm{supp}(K^-)$ respectively such that*

$$\Omega_m^+ = -\Omega_m^+ \ \textit{and} \ \Omega_j^- = -\Omega_j^- \ \textit{for all} \ m, j$$

$$\frac{1}{j} \cdot \Omega_j^- \cap U = \varnothing \ \textit{for all} \ j$$

$$\sum_{m=1}^{\infty} \psi_m^+(z) = 1 \ \textit{for all} \ z \in \operatorname{supp}(K^+) \setminus U$$

$$\sum_{j=1}^{\infty} \psi_j^-(z) = 1 \ \textit{for all} \ z \in \operatorname{supp}(K^-).$$

$$\psi_m^+(z) = \psi_m^+(-z) \ \textit{and} \ \psi_j^-(z) = \psi_j^-(-z) \ \textit{for all} \ m, j, z$$

Define

$$\begin{aligned}\tilde{K}(z) = K^+(z)\mathbf{1}_U(z) &+ \sum_{m=1}^{\infty} \frac{1}{m^{d+1}} K^+(z/m)\mathbf{1}_{\Omega_m^+}(z/m)\psi_m^+(z/m) \\ &+ \sum_{j=1}^{\infty} j^{d+1} K^-(jz)\mathbf{1}_{\Omega_j^-}(jz)\psi_j^-(jz).\end{aligned}$$

Then $\tilde{K}$ satisfies (1.16), $\tilde{K}$ is positive in a neighborhood of the origin, the energies satisfy the inequality $\mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}_\varepsilon) \geq \mathrm{HC}_\varepsilon(\mathbf{u}, \tilde{\boldsymbol{K}}_\varepsilon)$ and for every $n \in S^{d-1}$

$$\int_{\mathbb{R}^d} |z \cdot n| K(z) dz = \int_{\mathbb{R}^d} |z \cdot n| \tilde{K}(z)\, dz.$$

Proof. By design $\tilde{K}$ is strictly positive in a neighborhood of the origin and $\tilde{K}(z) = \tilde{K}(-z)$. As in the proof of Lemma 2.3.6 the equality $\int_{\mathbb{R}^d} |z \cdot n| K(z) dz = \int_{\mathbb{R}^d} |z \cdot n| \tilde{K}(z)\, dz$ will follow by changing variables and recollecting the various terms in the sums. Therefore we know $z\tilde{K}(z) \in L^1(\mathbb{R}^d)$. Since $\tilde{K}$ and K are equal on U it also follows that $\tilde{K} \in L^1(\mathbb{R}^d)$.

Again we use the fact that the energy is linear in the kernel to write

$$\begin{aligned}\mathrm{HC}_\varepsilon(\mathbf{u}, \tilde{\boldsymbol{K}}_\varepsilon) = \mathrm{HC}\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_U)_\varepsilon\big) &+ \sum_{m=1}^{\infty} \frac{1}{m^{d+1}} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon)(z/m)\big) \\ &+ \sum_{j=1}^{\infty} j^{d+1} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^-\mathbf{1}_{\Omega_j^-}\psi_j^-)_\varepsilon(jz)\big)\end{aligned}$$

as well as

$$\mathrm{HC}_\varepsilon(\mathbf{u}, \boldsymbol{K}) = \mathrm{HC}\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_U)_\varepsilon\big) + \sum_{m=1}^{\infty} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon(z)\big) + \sum_{j=1}^{\infty} \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^-\mathbf{1}_{\Omega_j^-}\psi_j^-)_\varepsilon(z)\big).$$

Thus we only need to show

$$\frac{1}{m^{d+1}}\mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon)(z/m)\big) \le \mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon(z)\big)$$

and

$$j^{d+1}\mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^-\mathbf{1}_{\Omega_j^-}\psi_j^-)_\varepsilon(jz)\big) \le \mathrm{HC}_\varepsilon\big(\mathbf{u}(\boldsymbol{K}^-\mathbf{1}_{\Omega_j^-}\psi_j^-)_\varepsilon(z)\big).$$

The second inequality follows from an argument identical to the one in Lemma 2.3.6. For the first inequality we can use Lemma 2.3.4 in the opposite direction to get

$$\frac{1}{m^{d+1}}\mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon)(z/m)\big) \le \frac{1}{m^{d+1}}\mathrm{HC}_{\varepsilon/m}\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_{\varepsilon/m})(z/m)\big).$$

Simplifying the right hand side of the above equation we see that it is indeed $\mathrm{HC}_\varepsilon\big(\mathbf{u}, (\boldsymbol{K}^+\mathbf{1}_{\Omega_m^+}\psi_m^+)_\varepsilon(z)\big)$. Thus the argument is complete.

□

The discussion preceding Lemma 2.3.7 proves the following more general version of Theorem 2.3.1

Theorem 2.3.8. *Suppose that each kernel K_{ij} can be written as a positive linear combination of m kernels $K_{ij}(z) = \sum_{r=1}^m \sigma_{i,j}^r K^r(z)$ where each K^r satisfies (1.16) and the constants $\sigma_{ij}^r > 0$ satisfy the triangle inequality (1.13) for each r. Further, suppose that for each K^r there is some rearrangement described in Lemma 2.3.7 such that the resulting $\tilde{K}^r$ is nonnegative. If we define*

$$\sigma_{i,j}(n) = \sum_{r=1}^m \sigma_{ij}^r \int_{\mathbb{R}^d} K^r(z)|z \cdot n|\, dz$$

then as $\varepsilon \to 0$ the Lyapunov functional $HC_\varepsilon(\cdot, \boldsymbol{K}_\varepsilon)$ Gamma converges in the L^1 topology over $\mathcal{K}$ to the energy $E(\cdot, \boldsymbol{\sigma})$ given in (2.45). Furthermore if for some sequence $\mathbf{u}_\varepsilon$ we have $\sup_{\varepsilon>0} HC_\varepsilon(\mathbf{u}_\varepsilon, \boldsymbol{K}_\varepsilon) < \infty$ then $\mathbf{u}_\varepsilon$ is pre-compact in $L^1(D)$ and the set of accumulation points is contained in $BV_{\mathcal{B}}(D)$.

In practice it is difficult to check whether a given kernel K has a nonnegative rearrangement $\tilde{K}$. However the following proposition gives a very simple necessary condition.

Proposition 2.3.9. *Suppose that some rearrangement of K produces a nonnegative $\tilde{K}$. Then for every $s \in (-\infty, 1]$ and every $X \subset \mathbb{R}^d$ that is star shaped with respect to the origin the integral $\int_X |x|^s K(x)\,dx$ is nonnegative.*

Proof. For $t \in \mathbb{R}^+$ let $t \cdot X = \{x \in \mathbb{R}^d : x/t \in X\}$. Then since X is star shaped with respect to the origin $t \cdot X \subset X$ for $t < 1$ and $X \subset t \cdot X$ for $t > 1$.

$$\begin{aligned}
\int_X |x|^s K(x)\,dx =& \sum_{j=1}^{\infty} \int_X |x|^s K(x) \mathbf{1}_{\Omega_j^+}(x) \psi_j^+(x)\,dx \\
&- \sum_{m=1}^{\infty} \int_X |x|^s |K(x)| \mathbf{1}_{\Omega_m^-}(x) \psi_m^-(x)\,dx \\
=& \sum_{j=1}^{\infty} \frac{1}{j^{s+d}} \int_{j\cdot X} |x|^s K(x/j) \mathbf{1}_{\Omega_j^+}(x/j) \psi_j^+(x/j)\,dx \\
&- \sum_{m=1}^{\infty} m^{s+d} \int_{\frac{1}{m}\cdot X} |x|^s |K(mx)| \mathbf{1}_{\Omega_m^-}(mx) \psi_m^-(mx)\,dx \\
\geq& \int_X |x|^s \tilde{K}(x)\,dx \geq 0
\end{aligned}$$

□

2.4 Conclusions

Recent developments in our understanding of threshold dynamics entail various assumptions on the convolution kernel used. However, much of the theory can be extended to a greater variety of kernels; the analysis presented in this chapter provides several examples. The added flexibility in the choice of the kernel is significant in applications ranging from machine learning to materials science. Moreover, restrictions can often be sidestepped by alternate forms of the basic algorithm that maintain its most beneficial qualities. Several such variants were introduced and rigorously studied in this chapter.

CHAPTER 3

Kernels with Prescribed Surface Tension and Mobility for Threshold Dynamics Schemes [2]

3.1 Introduction

The question of whether the original threshold dynamics algorithm, Algorithm 1.1, can be extended to anisotropic curvature flows, by appropriate choice of the convolution kernel K in (1.14), has been the topic of numerous investigations in the literature; a summary is given in Section 3.3. The present chapter is devoted to providing a decisive, constructive answer to this question, by showing how to choose the kernel K given a desired possibly anisotropic surface tension and possibly anisotropic mobility for the interface. In addition we give a complete characterization of surface tension and mobility pairs for which a positive kernel construction is possible. Combined with the new single-growth versions of threshold dynamics proposed in Chapter 2, the kernel constructions of this chapter yield unconditionally stable schemes for the weighted mean curvature flow of a general N-phase network by allowing N-choose-2 anistropic surface tensions and N-choose-2 anisotropic mobilities (one pair for each interface in the network) to be individually specified. This full level of generality is a first for threshold dynamics schemes.

3.2 Preliminaries and Notation

Given a set $\Omega \subset \mathbb{R}^d$, its *support function* h_Ω is defined to be

$$h_\Omega(x) = \sup_{y \in \Omega} x \cdot y. \tag{3.1}$$

[2] Joint work with Selim Esedoḡlu and Pengbo Zhang [29]. Published in *The Journal of Computational Physics*.

Note the simple but useful fact

$$h_{\Omega_1+\Omega_2} = h_{\Omega_1} + h_{\Omega_2} \tag{3.2}$$

where $\Omega_1 + \Omega_2$ denotes the *Minkowski sum* of the sets Ω_1 and Ω_2:

$$\Omega_1 + \Omega_2 = \{x \,:\, x = x_1 + x_2 \text{ with } x_1 \in \Omega_1 \text{ and } x_2 \in \Omega_2\} \tag{3.3}$$

i.e. it is just the *dilation* of Ω_1 by Ω_2.

We will denote the *spherical Radon transform* of an even function $f : \mathbb{S}^{d-1} \to \mathbb{R}$ by

$$\mathcal{J}_s f(n) = \int_{\mathbb{S}^{d-1}\cap n^\perp} f(x)\, dH^{d-1}(x). \tag{3.4}$$

Closely connected with the spherical Radon transform $\mathcal{J}_s$ is the *cosine transform* $\mathcal{T}$, also defined on even functions on the sphere, as follows:

$$\mathcal{T} f(n) = \int_{\mathbb{S}^{d-1}} f(x)\, |x \cdot n|\; dH^{d-1}(x). \tag{3.5}$$

The relation between $\mathcal{J}_s$ and $\mathcal{T}$ is given by

$$\Box\mathcal{T} = \mathcal{T}\Box = \mathcal{J}_s \tag{3.6}$$

where $\Box = \Delta_{\mathbb{S}^{d-1}} - (d-1)I$ and $\Delta_{\mathbb{S}^{d-1}}$ denotes the surface Laplacian (i.e. the Laplace-Beltrami operator) on $\mathbb{S}^{d-1}$. The operators $\mathcal{J}_s$, $\mathcal{T}$, and of course $\Box$ are symmetric, in the sense that

$$\int_{\mathbb{S}^{d-1}} \phi\mathcal{L}\psi\, dH^{d-1} = \int_{\mathbb{S}^{d-1}} \psi\mathcal{L}\phi\, dH^{d-1} \tag{3.7}$$

for $\mathcal{L} \in \{\mathcal{J}_s\,, \mathcal{T}\,, \Box\}$, where ϕ and ψ are any two smooth functions on $\mathbb{S}^{d-1}$. They also commute with one another. For a given anisotropy σ with a strongly convex B_σ and smooth ∂B_σ, its *generating function* $\omega : \mathbb{S}^{d-1} \to \mathbb{R}$ is defined to be the following even function:

$$\omega := \mathcal{T}^{-1}\sigma = \Box\mathcal{J}_s^{-1}\sigma = \mathcal{J}_s^{-1}\Box\sigma. \tag{3.8}$$

For $d = 2$, the expression (3.8) for the generating function of an anisotropy simplifies to

$$\omega(\theta) = \frac{1}{4}\left\{\sigma''\left(\theta - \frac{\pi}{2}\right) + \sigma\left(\theta - \frac{\pi}{2}\right)\right\} \tag{3.9}$$

where θ denotes the polar angle of an $x \in \mathbb{S}^1$; see e.g. [25]. We will use the following

definition of the Fourier transform on $\mathbb{R}^d$:

$$\widehat{f}(\xi) = \int_{\mathbb{R}^d} f(x) e^{-ix\cdot\xi}\, d\xi \text{ so that } f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \widehat{f}(\xi) e^{i\xi\cdot x}\, d\xi$$

for e.g. f in Schwartz class.

Background on the transforms (3.4) and (3.5) quoted above and their significance in convex geometry can be found in e.g. [39], [14], [43].

3.3 Previous Work

Generalizations of threshold dynamics to anisotropic surface energies have been considered in a number of works in the existing literature. The basic idea is to take the convolution kernel K to be a more general kernel (than the Gaussian) satisfying the properties (1.16) and (1.17).

One of the first contributions to the study of Algorithm 1.1 with general convolution kernels is the convergence result of Ishii, Pires, and Souganidis [48]. Recall that they established the convergence of the algorithm to the viscosity solution of the equation

$$u_t = F(D^2u, Du) \tag{3.10}$$

where

$$F(M,p) = \left(\int_{p^\perp} K(x)\, dH^{d-1}(x) \right)^{-1} \left(\frac{1}{2} \int_{p^\perp} \langle Mx\,,\, x\rangle K(x) dH^{d-1}(x) \right) \tag{3.11}$$

for $p \in \mathbb{R}^d$ and M a $d \times d$ symmetric matrix, provided that K is a *positive* convolution kernel with certain additional decay and continuity properties. Positivity of the kernel is required for the scheme to preserve the comparison principle that applies to underlying interfacial motion, and is essential for the viscosity solutions approach taken in [48]. On the other hand, the consistency calculation given in the paper applies to more general (e.g. sign changing) kernels (and also appears in [74] for the special case of a Gaussian). This study extends to the case of anisotropic curvature motions earlier proofs of convergence appearing in [30] and [4] for the isotropic version of the scheme that uses the Gaussian as the convolution kernel.

The paper by Ishii et. al. does not address the inverse problem of constructing a convolution kernel for a given surface tension (and possibly a mobility), which is perhaps the more practical problem from a numerical methods perspective. The first contribution in

this direction is by Ruuth & Merriman in [75], who propose a construction in $2D$. Given an anisotropy $f : [0, 2\pi] \to \mathbb{R}^+$, the focus of the authors is to construct a kernel (characteristic function of a judiciously chosen star shaped domain) that, when used in threshold dynamics, would generate a normal speed of the form

$$v_\perp(x) = (f''(\theta) + f(\theta))\,\kappa(x) \tag{3.12}$$

where θ is the angle that the unit outer normal $n(x)$ at x makes with the positive x-axis. This approach conflates the contributions to $v_\perp$ in (1.9) from mobility and surface tension factors. Indeed, there are infinitely many surface tension & mobility pairs (σ, μ) that correspond to the same f and hence the same normal speed in (3.12); the discussion in [75] does not elucidate the two factors. This is a particularly important matter in multiphase flows, since surface tensions determine the equilibrium condition at junctions according to (1.11). And in fact, it turns out that for Ruuth & Merriman's construction, the corresponding surface tension is *not* given by f in (3.12); see [25] for a detailed discussion. On the plus side, these kernels are positive (being characteristic functions) and thus preserve the comparison principle.

More recently, Bonnetier et. al. [15] have proposed a construction that works in both 2D and 3D. The Fourier transform of their kernels is explicit in terms of the surface tension:

$$\widehat{K}(\xi) = \exp\left(-\sigma^2(\xi)\right). \tag{3.13}$$

It turns out that the corresponding mobility satisfies $\mu := \sigma$ identically, see [25]. This construction often yields non-positive kernels, even in two dimensions, preventing the authors from giving a rigorous proof of convergence. Moreover, as soon as the anisotropy σ does not have an ellipsoid as its unit ball, (3.13) has a singularity at the origin, leading to slow decay of K. We will revisit this construction in Section 3.4.2 and remedy some of its shortcomings.

Finally, we introduce and recall the key ingredients for our kernel constructions. Firstly, recall the formulas obtained in [25] which give expressions for the surface tension σ_K and mobility μ_K corresponding to a convolution kernel K. These formulas are of central importance to this chapter so we reproduce them here:

$$\sigma_K(n) := \frac{1}{2}\int_{\mathbb{R}^d} |n \cdot x| K(x)\,dx. \tag{3.14}$$

$$\frac{1}{\mu_K(n)} := \int_{n^\perp} K(x)\,dH^{d-1}(x). \tag{3.15}$$

In polar coordinates, the expression for the surface tension σ_K is:

$$\sigma_K(n) = \frac{1}{2}\int_{\mathbb{S}^{d-1}} |n \cdot x| \int_0^\infty r^d K(rx)\, dr\, dH^{d-1}(x). \tag{3.16}$$

We should also note, as is done in [25], that equations (3.8) & (3.16) imply the generating function ω_K of the anisotropy σ_K that corresponds to a given kernel K is given by:

$$\omega_K(n) := \frac{1}{2}\int_0^\infty K(rn) r^d\, dr. \tag{3.17}$$

Equation (3.15) can alternatively be written using the spherical Radon transform $\mathcal{J}_s$:

$$\frac{1}{\mu_K} = \mathcal{J}_s \int_0^\infty K(rn) r^{d-2}\, dr. \tag{3.18}$$

Also in [25], the following alternative expressions for σ_K and μ_K in terms of the Fourier transform $\widehat{K}$ of the convolution kernel K are provided:

$$\begin{aligned} \sigma_K(n) &= -\frac{1}{2\pi}\ \text{F. P.} \int_{\mathbb{R}} \frac{\widehat{K}(n\xi)}{\xi^2}\, d\xi, \text{ and} \\ \mu_K(n) &= 2\pi \left(\int_{\mathbb{R}} \widehat{K}(n\xi)\, d\xi \right)^{-1}. \end{aligned} \tag{3.19}$$

Next we recall the *Barrier Theorem*, Theorem 1.1.3, from [25] that places a restriction on the positivity of convolution kernels in terms of the Wulff shape W_σ of the given anisotropy. Namely, σ_K can be the surface tension corresponding to a positive kernel K if and only if W_σ is a zonoid (refer to the discussion following Theorem 1.1.3 for a brief discussion of zonoids). As a result, our positive kernel constructions must be limited to cases where the corresponding surface tension σ_K is a zonoid.

The final key ingredient for our positive kernel constructions are the single growth variants of the MBO scheme, Algorithms 2.1, 2.3, and 2.4 from Chapter 2. The single growth algorithms guarantee unconditional stability for a much wider class of convolution kernels compared to the classical algorithm, Algorithm 1.1 and recent extensions Algorithms 1.2 and 2.2. In particular, positivity of the convolution kernel is sufficient to guarantee unconditional stability, there is no need for positivity in the Fourier domain. This allows us to improve upon the kernel constructions in [25], which built positivity in both the physical and Fourier domain, and thus could not specify mobilities.

3.4 The New Convolution Kernels

In this main section of the chapter, we present two new constructions of a convolution kernel K for a given, possibly anisotropic, target surface tension $\sigma_* : \mathbb{S}^{d-1} \to \mathbb{R}^+$ and target mobility $\mu_* : \mathbb{S}^{d-1} \to \mathbb{R}^+$. Both two and three dimensions are addressed. Both constructions identify the mobility and surface tension factors, and are therefore suitable for use in the multiphase setting.

The first construction, presented in Section 3.4.1, yields smooth, compactly supported kernels that are *positive*, so that Algorithm 1.1 preserves the monotonicity (comparison principle) of the underlying evolution (1.8). As already implied by the barrier Theorem 1.1.3, there is necessarily a difference between two and three dimensions in this endeavor. Our results give a fairly complete picture of when this goal can be accomplished, and how to do it. These kernels do not necessarily have positive Fourier transforms, but the variant, Algorithm 2.1, of threshold dynamics in the two-phase, and Algorithm 2.3 in the multiphase setting, ensure dissipation of the corresponding energy (1.18) and (1.21).

The second construction, presented in Section 3.4.2, is a variant of the construction of [15] and is the more general: It allows any convex surface tension σ_* and any positive mobility μ_*, both in two and three dimensions, and yields a convolution kernel K the Fourier transform $\widehat{K}$ of which is positive: $\widehat{K} \geq 0$. Moreover, unlike in [15], the resulting kernel is always Schwartz class. The original threshold dynamics scheme, Algorithm 1.1, using such a kernel thus dissipates the non-local interfacial energy (1.18) according to Proposition 1.1.1. However, as in [15], these kernels may not be positive, even in two dimensions.

3.4.1 Positive Kernels

In this section, we present new, *positive* convolution kernels for possibly anisotropic, desired surface tension and mobility pairs (σ_*, μ_*), leading to *monotone* threshold dynamics schemes. There are significant differences in two and three dimensions, so these two cases are discussed separately below. Proposition 3.4.2 in Section 3.4.1.1 yields positive, compactly supported, and smooth convolution kernels for essentially *any* anisotropic surface tension and mobility pair in two dimensions. In three dimensions, Lemma 3.4.3 and Proposition 3.4.5 of Section 3.4.1.2 essentially classify all anisotropic surface tension and mobility pairs for which such a kernel can be found, and for all such cases exhibit the desired kernels explicitly. Let us summarize the implication of these constructions by the following immediate consequence of results from [48] and [25]:

Corollary 3.4.1. *In two dimensions, given essentially any anisotropic surface tension and mobility pair, a convolution kernel can be found such that two-phase threshold dynamics algorithms, Algorithms 1.1 and 2.1, when extended to bounded uniformly continuous functions in the standard manner, converge to the viscosity solution of the corresponding evolution (1.8) on any finite time interval. Proposition 3.4.2 exhibits these kernels.*

In three dimensions, there are surface tension and mobility pairs for which a monotone threshold dynamics scheme cannot be constructed. For essentially all those for which it can, Proposition 3.4.5 exhibits kernels with which the extension of Algorithms 1.1 and 2.1 to uniformly continuous functions will again converge to the viscosity solution of the corresponding evolution.

Our approach is as follows: Given (σ_*, μ_*), according to (3.14) and (3.15), we will need to solve the following coupled system of integral equations:

$$\begin{cases} \displaystyle\int_{\mathbb{R}^d} K(x)|n \cdot x|\, dx = \sigma_*(n), \text{ and} \\ \displaystyle\int_{n^\perp} K(x)\, dH^{d-1}(x) = \frac{1}{\mu_*(n)} \end{cases} \tag{3.20}$$

for the unknown function K. Note that there is in fact vast non-uniqueness of the solution, so it may be possible to impose additional conditions (besides positivity).

System (3.20) can be more conveniently expressed using the cosine and spherical Radon transforms as in formulas (3.17) & (3.18):

$$\begin{cases} \displaystyle\int_0^\infty K(rn) r^d\, dr = \omega_*(n) := \mathcal{T}^{-1}\sigma_*(n), \text{ and} \\ \displaystyle\int_0^\infty K(rn) r^{d-2}\, dr = \mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right](n). \end{cases} \tag{3.21}$$

For both $d = 2$ and $d = 3$, the essential idea is the following: Formulas (3.14) & (3.15) indicate that surface tension and mobility of a kernel vary differently as the convolution kernel is dilated along radial directions. We exploit this simple observation. With that in mind, let $\eta : \mathbb{R} \to \mathbb{R}$ be the following compactly supported, smooth, positive bump function:

$$\eta(x) = \begin{cases} \exp\left(-\dfrac{1}{x^2(x-2)^2}\right) & \text{if } x \in (0,2), \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

For $j \in \mathbb{N}^+$, let

$$m_j = \int_0^2 x^j \eta(x)\, dx \tag{3.23}$$

denote its moments.

3.4.1.1 Positive Kernels in 2D

In two dimensions, it turns out that a *positive*, smooth, compactly supported convolution kernel that is strictly positive at the origin can be constructed for *any* given surface tension & mobility pair (σ_*, μ_*) that satisfies the following minimal assumptions:

(1.1) B_{σ_*} is strongly convex and ∂B_{σ_*} is smooth,

(1.2) $\mu_* : \mathbb{S}^1 \to \mathbb{R}^+ \setminus \{0\}$ is smooth.

We have

Proposition 3.4.2. *Under conditions* (1.1) *&* (1.2) *on σ_* and μ_*, there exists a positive, smooth, compactly supported convolution kernel $K : \mathbb{R}^2 \to \mathbb{R}^+$ such that $\sigma_K = \sigma_*$ and $\mu_K = \mu_*$.*

Proof: To solve the system (3.20), we look for a kernel K that in polar coordinates has the form

$$K(r, \theta) = \alpha(\theta)\eta\left(\beta(\theta)r\right) \tag{3.24}$$

where $\alpha, \beta : \mathbb{R} \to \mathbb{R}^+$ are π-periodic, smooth functions. Substituting (3.24) into (3.21) gives

$$\begin{aligned} \frac{\alpha(\theta)}{\beta^3(\theta)} m_2 &= \omega_*(\theta), \text{ and} \\ \frac{\alpha(\theta)}{\beta(\theta)} m_0 &= \mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right](\theta). \end{aligned} \tag{3.25}$$

Note that for $d = 2$,

$$\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right](\theta) = \frac{1}{\mu_*\left(\theta - \frac{\pi}{2}\right)} \tag{3.26}$$

and by (3.9),

$$\omega_*(\theta) = \frac{1}{4}\left\{\sigma_*''\left(\theta - \frac{\pi}{2}\right) + \sigma_*\left(\theta - \frac{\pi}{2}\right)\right\}. \tag{3.27}$$

Note that by our assumption on B_{σ_*} above, $\omega_*(\theta) > 0$ on $\theta \in [0, 2\pi]$. Solving system

(3.25) for α and β and using (3.26) & (3.27) gives

$$\begin{aligned}\alpha(\theta) &= \left(\frac{4m_2}{m_0^3\mu_*^3\left(\theta-\frac{\pi}{2}\right)\left[\sigma_*''\left(\theta-\frac{\pi}{2}\right)+\sigma_*\left(\theta-\frac{\pi}{2}\right)\right]}\right)^{\frac{1}{2}} \\ \beta(\theta) &= \left(\frac{4m_2}{m_0\mu_*\left(\theta-\frac{\pi}{2}\right)\left[\sigma_*''\left(\theta-\frac{\pi}{2}\right)+\sigma_*\left(\theta-\frac{\pi}{2}\right)\right]}\right)^{\frac{1}{2}}\end{aligned} \tag{3.28}$$

Formulas (3.24) & (3.28) give an explicit prescription for the convolution kernel in terms of the desired surface tension σ_* and mobility μ_*. □

Remark: The kernel can be easily modified to be strictly positive at the origin: Replace (σ_*, μ_*) in the construction with $(\tilde{\sigma}_*, \tilde{\mu}_*)$ where

$$\tilde{\sigma}_*(x) = \sigma(x) - \varepsilon \text{ and } \frac{1}{\tilde{\mu}_*(x)} = \frac{1}{\mu_*(x)} - \varepsilon$$

and $\varepsilon > 0$ is chosen small enough so that $B_{\tilde{\sigma}}$ is strongly convex and $\tilde{\mu}_* > 0$. Denote the resulting kernel $\tilde{K}$. Then, the kernel $K = \tilde{K} + \exp\left(-\varepsilon|x|^2\right)$ satisfies $\sigma_K = \sigma_*$ and $\mu_K = \mu_*$. □

3.4.1.2 Positive Kernels in 3D

The situation is more complicated in three dimensions. The essential question is positivity of the right hand sides of the system of integral equations (3.21) that entail the inverse cosine and inverse spherical Radon transforms. It turns out that such positivity questions are intimately connected with long studied problems and certain mathematical objects arising in convex geometry. This connection with convex geometry literature was already noted and utilized in [25] in formulating its barrier Theorem 1.1.3 quoted in Section 3.3. That theorem says that a necessary condition for a positive convolution kernel to be found for the target anisotropy $\sigma_* : \mathbb{S}^2 \to \mathbb{R}^+$ (regardless of the mobility) is that the corresponding Wulff shape W_{σ_*} be a *zonoid* (also known as a *projection body*), an important class of centrally symmetric convex bodies that appear prominently in the convex geometry literature; see Section 1.1.2 for a brief discussion and e.g. [14, 39] for much more.

When we confront the question of simultaneously achieving both a target zonoidal surface tension σ_* *and* a target mobility μ_* with a positive convolution kernel, a related class of objects, known as *intersection bodies* [56], and their connections to a widely studied problem known as the *Busemann-Petty problem* [18], again from convex geometry, come into play. In what follows, we will need to appeal to the resolution of this problem in [35]

for the case $d = 3$.

Our first result in this direction is the barrier type Lemma 3.4.3 below. It states that in order to accommodate a wide enough class of mobilities μ_* using positive convolution kernels, we need to demand more from W_{σ_*} than just being a zonoid. The issue is that there are strongly convex and smooth zonoids the generating functions ω_* of which vanish somewhere on $\mathbb{S}^2$. If ω_* corresponding to σ_* vanishes even at a single point, however, it turns out μ_* cannot arise from the gradient descent formulation (1.10) of the interfacial motion (1.8):

Lemma 3.4.3. *Let $\sigma_* : \mathbb{R}^3 \to \mathbb{R}^+$ be an anisotropy such that W_{σ_*} is a smooth and strongly convex zonoid. If K is a positive convolution kernel such that $\sigma_K = \sigma_*$ and its corresponding mobility $\mu_K : \mathbb{S}^2 \to \mathbb{R}^+ \setminus \{0\}$ is smooth with a convex one-homogeneous extension to $\mathbb{R}^3$, then W_{σ_*} can be written as the Minkowski sum of a zonoid and a sphere. In particular, a threshold dynamics scheme that is consistent with an evolution law (1.8) arising from the gradient descent formulation (1.10) cannot possibly be monotone unless W_{σ_*} is the dilation of a zonoid by a sphere.*

To prove Lemma 3.4.3 we need a slightly stronger version of Gardner's intersection body result in [35]. Gardner showed that the radial function of a centered convex body (a compact convex set that is centrally symmetric about the origin) has non-negative inverse spherical Radon transform, however upon close inspection of Theorem 5.2 and its Corollary 5.3 in [35] we note that in fact the inverse spherical Radon transform is strictly positive. Before we can give our argument, we need to introduce the concept of the *Schwarz symmetral* of a convex body, which is essential to Gardner's approach.

Definition: The Schwarz symmetral $\bar{K} \subset \mathbb{R}^d$ of a convex body $K \subset \mathbb{R}^d$ about the axis $n \in \mathbb{S}^{d-1}$ is a convex body of revolution with respect to the axis n. $\bar{K}$ is constructed from K as follows. Let l_n be the line parallel to n through the origin. For each $x \in l_n$ let D_x be the (possibly degenerate) $(d-1)$-dimensional ball centered at x contained in the hyperplane $x + n^\perp$ with $(d-1)$-dimensional Hausdorff measure equal to $H^{d-1}\big(K \cap (x + n^\perp)\big)$. Then the Schwarz symmetral is given by $\bar{K} = \bigcup_{x \in l_n} D_x$. Clearly, $\bar{K}$ and K have the same d-dimensional volume, and it follows from the Brunn-Minkowski Theorem that $\bar{K}$ is a convex body of revolution about the axis n. If in addition K is a centered convex body, then it follows that the $(d-1)$-dimensional volume of the slices $H^{d-1}\big(\bar{K} \cap (x + n^\perp)\big) = H^{d-1}\big(K \cap (x + n^\perp)\big)$ is decreasing in $|x|$.

Lemma 3.4.4. *If $\mu : \mathbb{S}^2 \to \mathbb{R}^+$ is an even smooth function with a convex one-homogenous extension to $\mathbb{R}^3$ then*

$$\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right] > 0 \tag{3.29}$$

on $\mathbb{S}^2$.

Proof. Let ρ be the radial function of the unit ball of μ. Many of the results in [35] require strict convexity of the radial functions. As a result, we introduce a spherical perturbation of μ. Let $\mu_\varepsilon(x) = \mu(x) + \varepsilon|x|$. Then μ_ε is smooth and strictly convex. Let ρ_ε be the radial function of the unit ball μ_ε then

$$\rho_\varepsilon(x) = \frac{1}{\mu(x) + \varepsilon|x|} = \rho(x) \sum_{k=0}^{\infty} \varepsilon^k (-\frac{|x|}{\mu(x)})^k.$$

μ is bounded away from zero on the sphere thus we know that $\frac{|x|}{\mu(x)}$ is a smooth function on the sphere.

Let $g_\varepsilon, g \in C_e^\infty(\mathbb{S}^2)$ be the inverse spherical Radon transforms of ρ_ε and ρ respectively. Both the spherical Laplacian and the inverse Radon transform can be written as multiplication operators over the space of even Laplace spherical harmonics. Letting Y_{2n}^k be the $(k, 2n)^{\text{th}}$ Laplace spherical harmonic, we have from [32]

$$\mathcal{J}_s^{-1} Y_{2n}^k = Y_{2n}^k (-1)^n \prod_{j=1}^{n} \frac{2j}{2j-1},$$

$$\Delta_{\mathbb{S}^2} Y_{2n}^k = 2n(2n+1) Y_{2n}^k.$$

A simple estimate gives

$$\prod_{j=1}^{n} \frac{2j}{2j-1} < e\sqrt{2n}$$

The spherical harmonics are an orthonormal basis for $L^2(\mathbb{S}^2)$ thus comparing the multiplier coefficients it follows that there exists $B > 0$ such that for any $h \in C^2(\mathbb{S}^2)$

$$B(||\Delta_{\mathbb{S}^2} h||_{L^2} + ||h||_{L^2}) \geq ||\mathcal{J}_s^{-1} h||_{L^2}.$$

Clearly ρ_ε converges to ρ uniformly. Applying the spherical Laplacian to ρ_ε it also follows that $\Delta_{\mathbb{S}^2}\rho_\varepsilon \to \Delta_{\mathbb{S}^2}\rho$ uniformly. Using the above inequality we may then conclude that $g_\varepsilon \to g$ in $L^2(\mathbb{S}^2)$.

Choose some $n \in \mathbb{S}^2$ and let $\bar{K}_\varepsilon$ and $\bar{K}$ be the Schwarz symmetrals of the unit ball of μ_ε and μ respectively about the axis n. Let $\rho_{\bar{K}_\varepsilon}$ and $\rho_{\bar{K}}$ be the radial functions of $\bar{K}_\varepsilon$ and $\bar{K}$. If $\bar{g}_\varepsilon$ are the inverse spherical Radon transforms of $\rho_{\bar{K}_\varepsilon}$ then $g_\varepsilon(n) = \bar{g}_\varepsilon(n)$ by Theorem 4.1 and 5.2 in [35].

Let ϕ be the angle formed with the n-axis by drawing a line from the origin to a point on the boundary of $\bar{K}$. Since $\bar{K}$ and $\bar{K}_\varepsilon$ have rotational symmetry about the n-axis we may represent $\rho_{\bar{K}_\varepsilon}$ and $\rho_{\bar{K}}$ as functions of the angle ϕ. By the convexity and central symmetry of μ we know that $\sin(\phi)\rho_{\bar{K}}(\phi)$ is non-decreasing in ϕ for $\phi \in [0, \pi/2]$. By the continuity of $\rho_{\bar{K}}$ there must exist $0 < \phi_1 < \phi_2 \leq \pi/2$ so that $\sin(\phi_1)\rho_{\bar{K}}(\phi_1) < \sin(\phi_2)\rho_{\bar{K}}(\phi_2)$ since the function is 0 for $\phi = 0$ and positive for $\phi = \pi/2$.

By the inversion formula in [35] we may write

$$g_\varepsilon(n) = \bar{g}_\varepsilon(n) = \left(\frac{d}{dt} \int_0^1 \frac{st\rho_{\bar{K}_\varepsilon}(\arcsin(st))}{\sqrt{1-s^2}} ds \right)_{t=1}.$$

Write $f_\varepsilon(st) = st\rho_{\bar{K}_\varepsilon}(\arcsin(st))$. Theorem 5.2 in [35] shows that $f_\varepsilon \in C^1$ thus

$$g_\varepsilon(n) = \int_0^1 \frac{sf'_\varepsilon(s)}{\sqrt{1-s^2}} ds.$$

Since f_ε is non-decreasing we have the inequality

$$g_\varepsilon(n) \geq \sin(\phi_1) \int_{\sin(\phi_1)}^{\sin(\phi_2)} f'_\varepsilon(s) ds = \sin(\phi_1)(f_\varepsilon(\sin(\phi_2)) - f_\varepsilon(\sin(\phi_1)).$$

$\rho_{\bar{K}_\varepsilon}$ must approach $\rho_{\bar{K}}$ thus it follows that there exists some $\varepsilon_0(n)$ such that for all $\varepsilon < \varepsilon_0(n)$ we get

$$g_\varepsilon(n) > \frac{1}{2}\sin(\phi_1)(\sin(\phi_2)\rho_{\bar{K}}(\phi_2) - \sin(\phi_1)\rho_{\bar{K}}(\phi_1)) > 0.$$

The direction n was arbitrary and $\mathbb{S}^2$ is compact therefore there exists $\delta > 0$ and $\varepsilon_1 > 0$ such that for every $n \in \mathbb{S}^2$ and all $\varepsilon < \varepsilon_1$ we have $g_\varepsilon(n) > \delta$. Now the convergence of g_ε to g in $L^2(\mathbb{S}^2)$ implies that there exists a subsequence g_{ε_j} which converges pointwise almost everywhere to g. Thus g must be larger than δ almost everywhere. However g is continuous, thus g must be larger than δ everywhere. Therefore g is strictly positive and we are done. □

Now we can prove Lemma 3.4.3.

Proof of Lemma 3.4.3: By hypothesis, $\mu_K = \mu_*$ extends as a norm to $\mathbb{R}^3$. Therefore, $\frac{1}{\mu_*}$ is the radial function of a convex body in $\mathbb{R}^3$. The extension of Theorem 5.2 and its Corollary 5.3 in [35] by Lemma 3.4.4, gives us

$$\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right] > 0 \tag{3.30}$$

on $\mathbb{S}^2$. Then, (3.18) implies that

$$\int_0^\infty K(rx)r\,dr > 0 \text{ for all } x \in \mathbb{S}^2. \tag{3.31}$$

But then, we also have

$$\omega_*(x) = \omega_K(x) = \int_0^\infty K(rx)r^3\,dr > 0 \text{ for all } x \in \mathbb{S}^2. \tag{3.32}$$

Let $\varepsilon = \frac{1}{2}\min_{x\in\mathbb{S}^2}\omega_*(x) > 0$, and define the new anisotropy

$$\sigma'_*(n) = \int_{\mathbb{S}^2} (\omega_*(x) - \varepsilon)\,|x \cdot n|\,dH^2(x). \tag{3.33}$$

Then, we can write

$$\sigma_*(x) = \sigma'_*(x) + \varepsilon|x| \tag{3.34}$$

so that by (3.2),

$$h_{W_{\sigma_*}} = h_{W_{\sigma'_*}} + h_{W_{\varepsilon|x|}}. \tag{3.35}$$

That implies

$$W_{\sigma_*} = W_{\sigma'_*} + W_{\varepsilon|x|}. \tag{3.36}$$

$W_{\sigma'_*}$ is a zonoid since its generating function is $\omega_* - \varepsilon \geq 0$, and $W_{\varepsilon|x|}$ is a sphere. □

Lemma 1 motivates placing the assumption on σ_* that W_{σ_*} be the dilation of a zonoid by a sphere, which we will adopt for the rest of this section. This is a dense subset of zonoids in the Hausdorff metric. As another difference of three dimensions from two, it turns out that even with this stronger assumption on σ_*, the mobility factor μ_* in a given target surface tension & mobility pair (σ_*, μ_*) needs to satisfy certain additional necessary conditions in order for there to exist a positive convolution kernel K with $\sigma_K = \sigma_*$ and $\mu_K = \mu_*$. Indeed, unlike for $d = 2$, not every positive function can arise as the mobility associated with a convolution kernel via equation (3.15), even if σ_* satisfies the conclusion of Lemma 3.4.3. This can be seen with the following example: Take μ_* to be

$$\mu_*(\theta, \phi) = \left(\mathcal{J}_s\left[1 - 2\eta\left(\frac{\phi - \frac{1}{2}}{\delta}\right)\right]\right)^{-1} \tag{3.37}$$

Choosing $\delta > 0$ small enough, we see that $\mu_*(n) > 0$ for all $n \in \mathbb{S}^2$. Assume that for some

$K \geq 0$, we have $\mu_K = \mu_*$. Then, by (3.18) and the injectivity of $\mathcal{J}_s$, we have

$$\int_0^\infty K(r,\theta,\phi) r^3 \, dr = 1 - 2\eta\left(\frac{\phi - \frac{1}{2}}{\delta}\right). \tag{3.38}$$

The left hand side is always positive, but the right hand side is negative for $\phi \approx 0$, which is a contradiction.

The good news is that for the large and natural class of *convex* mobilities, it is possible to construct positive convolution kernels, as long as the surface tension satisfies the conclusion of Lemma 3.4.3. In $d = 3$, we are thus led to the following assumptions on the pair (σ_*, μ_*):

(2.1) B_{σ_*} is strongly convex and ∂B_{σ_*} is smooth,

(2.2) W_{σ_*} is the dilation of a zonoid by a sphere,

(2.3) $\mu_* : \mathbb{S}^2 \to \mathbb{R}^+ \setminus \{0\}$ is smooth, and

(2.4) $\mu_* : \mathbb{R}^3 \to \mathbb{R}^+$ defined as $\mu_*(x) = |x|\mu_*\left(\frac{x}{|x|}\right)$ is convex.

It is worth repeating that condition (2.2) is essentially necessary, as explained above. It is also quite general, since it allows approximating any anisotropy that can arise as the continuum limit of a ferromagnetic Ising model; see e.g. the discussion in [25]. Furthermore, condition 4 is very natural: It merely stipulates that evolution (1.8) arises as gradient descent for the variational model (1.4) with respect to a possibly anisotropic *norm* on normal vector fields, as discussed in Section 3.2. We can now present our construction:

Proposition 3.4.5. *Under conditions* (2.1)-(2.4) *on* σ_* *and* μ_**, there exists a positive, smooth, compactly supported convolution kernel* $K : \mathbb{R}^3 \to \mathbb{R}^+$ *such that* $\sigma_K = \sigma_*$ *and* $\mu_K = \mu_*$.

Proof: As for $d = 2$, we look for a kernel K that in spherical coordinates has the form

$$K(r,\theta,\phi) = \alpha(\theta,\phi)\eta\left(\beta(\theta,\phi)r\right) \tag{3.39}$$

where $\alpha, \beta : \mathbb{R}^2 \to \mathbb{R}^+$ are 2π-periodic in each variable, smooth, and invariant under $(\theta,\phi) \to (\theta+\pi, \phi+\pi)$. Substituting (3.39) into (3.21) this time gives

$$\begin{aligned} &\frac{\alpha(\theta,\phi)}{\beta^4(\theta,\phi)} m_3 = \omega_*(\theta,\phi), \text{ and} \\ &\frac{\alpha(\theta,\phi)}{\beta^2(\theta,\phi)} m_1 = \mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right](\theta,\phi). \end{aligned} \tag{3.40}$$

Hypothesis on σ_* ensures that $\omega_*(\theta,\phi) > 0$. Thanks again (as in Lemma 3.4.3) to Theorem 5.2 of [35],

$$\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right] > 0. \tag{3.41}$$

since our hypothesis on μ_* implies that $\frac{1}{\mu_*}$ is the radial function of a convex body with smooth boundary.

Solving for $\alpha(\theta,\phi)$ and $\beta(\theta,\phi)$ in (3.40), we get

$$\begin{aligned}\alpha(\theta,\phi) &= \frac{m_3}{m_1^2}\frac{\left(\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right]\right)^2}{\mathcal{J}_s^{-1}\Box\sigma_*}, \text{ and} \\ \beta(\theta,\phi) &= \frac{m_3\mathcal{J}_s^{-1}\left[\frac{1}{\mu_*}\right]}{m_1\mathcal{J}_s^{-1}\Box\sigma_*}\end{aligned} \tag{3.42}$$

expressed in terms of standard transforms. Both are positive functions. Formulas (3.39) & (3.42) provide an explicit prescription for the desired convolution kernel, which is positive. □

Remark: In fact, in the language of e.g. [34, 35, 56], a mobility μ_* can arise from a positive convolution kernel in threshold dynamics algorithms if and only if $\frac{1}{\mu_*}$ is the radial function of an *intersection body*. However, this characterization is almost by definition of an intersection body (which is not as transparent as that of a zonoid), and therefore does not shed much light on the matter.

3.4.2 Kernels with Positive Fourier Transform

It turns out that we can construct a smooth, rapidly decaying convolution kernel with positive Fourier transform, in any spatial dimension d, as long as the target surface tension $\sigma_* : \mathbb{S}^{d-1} \to \mathbb{R}^+$ and the target mobility $\mu_* : \mathbb{S}^{d-1} \to \mathbb{R}^+$ satisfy the minimal assumptions (1.1) & (1.2) of Section 3.4.1.1. Our construction and resulting kernels are similar to those of [15], but are more general since we accommodate *any* mobility, whereas the kernels of [15] are restricted to the (important) special case $\mu_* = \sigma_*$. Moreover, the kernels of [15] are singular in the Fourier domain for all but ellipsoidal anisotropies, leading to slow decay in the physical domain. This technical issue is also easily rectified with our construction.

Recall that according to (3.19), we can find a kernel K with the desired target surface

tension and mobility by solving the system

$$\begin{cases} \text{F. P.} \displaystyle\int_{\mathbb{R}} \frac{\widehat{K}(n\xi)}{\xi^2}\, d\xi = -2\pi\sigma_*(n), \text{ and} \\ \displaystyle\int_{\mathbb{R}} \widehat{K}(n\xi)\, d\xi = \frac{1}{2\pi\mu_*(n)}. \end{cases} \tag{3.43}$$

This is particularly simple since the equations are pointwise in $n \in \mathbb{S}^{d-1}$ (unlike in the physical space construction of Section 3.4.1, as we shall see). We have:

Proposition 3.4.6. *Under conditions* (1.1) *&* (1.2) *on σ_* and μ_*, there exists a convolution kernel $K : \mathbb{R}^d \to \mathbb{R}$ in Schwartz class and a constant $\gamma > 0$ such that $\sigma_K = \sigma_*$, $\mu_K = \gamma\mu_*$, and $\widehat{K} \geq 0$.*

Proof Let $\zeta : \mathbb{R} \to \mathbb{R}$ be a smooth function that satisfies the following conditions:

1. $\zeta(x) \geq 0$ and $\zeta(x) = \zeta(-x)$ for all x,
2. $\zeta(x) = 0$ for $|x| \leq 1$,
3. $\zeta(x) = x^2$ for $|x| \geq 2$.

We want to construct a kernel K such that $\widehat{K}$ will satisfy (3.43). Let $\alpha, \beta : \mathbb{R}^d \to [0, \infty]$ be one homogeneous functions. Then define the kernel K by:

$$\widehat{K}(\xi) = \frac{1}{2}\exp\Big(-\zeta\big(\alpha(\xi)\big)\Big) + \frac{1}{2}\exp\Big(-\zeta\big(\beta(\xi)\big)\Big). \tag{3.44}$$

Substituting (3.44) into (3.43) yields the following equations on α and β in terms of the targets σ_* and μ_*:

$$\gamma\mu_*^{-1}(n) = \frac{1}{4\pi}\int_{\mathbb{R}} \exp\Big(-\zeta\big(\xi\alpha(n)\big)\Big) + \exp\Big(-\zeta\big(\xi\beta(n)\big)\Big) d\xi \tag{3.45}$$

and

$$\sigma_*(n) = \frac{1}{4\pi}\int_{\mathbb{R}} \frac{1-\exp\Big(-\zeta\big(\xi\alpha(n)\big)\Big)}{\xi^2} + \frac{1-\exp\Big(-\zeta\big(\xi\beta(n)\big)\Big)}{\xi^2} d\xi. \tag{3.46}$$

Note that we have introduced a constant γ to the mobility μ to ensure that a solution will exist. Let

$$s_0 = \frac{1}{4\pi}\int_{\mathbb{R}} e^{-\zeta(\xi)} d\xi \text{ and } s_2 = \frac{1}{4\pi}\int_{\mathbb{R}} \frac{1-e^{-\zeta(\xi)}}{\xi^2} d\xi.$$

Then, with the changes of variables $\xi \to \xi\alpha(n)$ and $\xi \to \xi\beta(n)$, equations (3.45) and (3.46) become

$$\frac{\gamma\mu_*^{-1}(n)}{s_0} = \frac{1}{\alpha(n)} + \frac{1}{\beta(n)}, \tag{3.47}$$

and

$$\frac{\sigma_*(n)}{s_2} = \alpha(n) + \beta(n). \tag{3.48}$$

To simplify notation let $a(n) = \frac{\gamma\mu_*^{-1}(n)}{s_0}$ and $b(n) = \frac{\sigma_*(n)}{s_2}$. Eliminating $\beta(n)$ in (3.47) & (3.48) and rearranging leads to the following quadratic in $\alpha(n)$:

$$\alpha(n)^2 - b(n)\alpha(n) + \frac{b(n)}{a(n)} = 0. \tag{3.49}$$

Solving for $\alpha(n)$ in (3.49) and then for $\beta(n)$ in (3.48), we get

$$\alpha(n) = \frac{1}{2s_2\gamma^{\frac{1}{2}}}\left(\gamma^{\frac{1}{2}}\sigma_* + \sqrt{\gamma\sigma_*^2 - 4s_0s_2\mu_*\sigma_*}\right) \tag{3.50}$$

$$\beta(n) = \frac{1}{2s_2\gamma^{\frac{1}{2}}}\left(\gamma^{\frac{1}{2}}\sigma_* - \sqrt{\gamma\sigma_*^2 - 4s_0s_2\mu_*\sigma_*}\right) \tag{3.51}$$

In order for both solutions to be real we need $b(n) \geq \frac{4}{a(n)}$. Therefore, we need to choose γ such that $\gamma \geq \frac{4s_0s_2\mu_*(n)}{\sigma_*(n)}$ for all n. We also do not want the discriminant to vanish, since this may introduce singularities into $\alpha(n), \beta(n)$ that are not present in the anisotropy and mobility. Indeed as long as the discriminant does not vanish α and β will have the same differentiability properties as μ_* and σ_*. However, at the same time, we would like $\alpha(n)$ and $\beta(n)$ to be as close to each other as possible so that the kernel is easy to sample. This suggests that a good choice for γ is

$$\gamma = (1+\varepsilon)\max_{n\in S^{d-1}} \frac{4s_0s_2\mu_*(n)}{\sigma_*(n)} \tag{3.52}$$

for some small $\varepsilon > 0$. Formulas (3.44), (3.47) & (3.48) provide an explicit prescription for the desired kernel K. □

As explained in [25], equations (3.20) along with the favorable smoothness and decay properties noted above imply that kernels (3.44) satisfy all the conditions for the *consistency* step of the convergence proof in [48]; we thus have the following as an immediate consequence:

Corollary 3.4.7. *For essentially any given anisotropic surface tension and mobility pair, whether in two or three dimensions, there exists a Schwartz class convolution kernel for*

which threshold dynamics algorithm, Algorithm 1.1, is consistent *with the evolution law (1.8).*

Although the sign changing character (in physical space) of the convolution kernels in this section (as well as in [15]) precludes an immediate proof of convergence (to the corresponding viscosity solution) in the mold of [48], numerical evidence does not indicate any adverse effects, at least in the two phase setting.

3.5 Numerical Evidence

We demonstrate the new kernel constructions of Section 3.4 on both two and multi-phase curvature flow problems, and in two-dimensional cases compare against front tracking simulations. To compare with front tracking whenever possible, we focus mostly on regular behavior (i.e. no topological changes), but of course, as is well known, threshold dynamics methods shine when it comes to gracefully handling topology changes. (Experiment of Figure 3.6 clearly indicates that this major benefit of threshold dynamics is completely preserved by the new algorithms and constructions of the present paper). Section 3.5.1 is devoted to experiments with two-phase anisotropic flows, while Section 3.5.2 focuses on multi-phase anisotropic flows. We demonstrate the original threshold dynamics algorithm, Algorithm 1.1, as well as its recent, fully anisotropic, multiphase extensions Algorithm 2.2 and Algorithm 2.3 that are recalled in the Appendix, using these kernels.

The front tracking algorithm used for two-dimensional comparisons represents the interfaces via explicit parametrizations. As such, it is essentially a finite differences discretization for a coupled system of parabolic PDEs in one space dimension, and can thus yield very accurate benchmark results by choosing a very fine discretization. (A parametric approach can be taken to models of this paper also in 3D, see e.g. [6]. Unfortunately, topological changes are a serious difficulty with this approach, especially in 3D, and a major motivation for methods that represent interfaces implicitly, such as that of this paper, or the phase field, or the level set method). We used explicit (forward Euler) time stepping to keep matters as simple as possible. In our implementation, triple junctions are common discretization points for the curves that meet at them. Their coordinates are updated not directly by the curvature flow PDE, but by iteratively optimizing the energy of the system with respect to these coordinates only, at every time step, much as described in [52]. This is how the Herring condition (1.11) is imposed at the junctions. As is well known, the parametrization of the curves can drift very far from an arc-length parametrization, resulting in very non-uniformly spaced points on the curves, adversely affecting the

stability (CFL) condition [17]. Hence, as is customary in front tracking, we periodically reparametrize the curves by arc-length, though as seldomly as possible to prevent accumulation of errors caused by small but inevitable perturbation to the curves during the process (an alternative is the approach of [47], or of e.g. [6] that also works in 3D, which maintain a regular parametrization through tangential velocities).

A few comments are in order regarding implementation of threshold dynamics algorithms as well. The consistency calculations (truncation error analysis) carried out e.g. in [48, 74, 25] reveal that in the two-phase setting, one would expect an error $O(\delta t)$ as $\delta t \to 0$. Similar calculations in [73] indicate that in the presence of junctions, the error becomes $O(\sqrt{\delta t})$, which is easy to understand: the kernels have width $\sqrt{\delta t}$ and hence start *sensing* the junction at any point of comparable distance on the smooth interfaces. Although these are modest convergence rates, they can be easily improved e.g. by Richardson extrapolation demonstrated in [72], [73] to be effective on threshold dynamics schemes, with or without junctions. Other important improvements include implementation on adaptive grids while maintaining the efficiency of Fourier transform based convolutions, also developed and demonstrated in [73] to achieve excellent accuracy.

Since our focus in this study is primarily on developing and verifying the requisite theory that enables adapting threshold dynamics to contexts in which *no version of it* so far exists due to a lack of fundamental understanding, we work with essentially the most basic version of the algorithms, except for the following well-known and very simple implementation detail to enhance spatial resolution: The convolution step arising in each threshold dynamics algorithm considered – such as (1.14) of the original MBO scheme, Algorithm 1.1 – yields a smooth level set function that can be used (via interpolation) to estimate what *fraction* of each grid cell is occupied by the evolving set at the next time step, which can then be used in representing the characteristic function of the set. The more involved improvements mentioned above, which are very important in practical applications of threshold dynamics, can of course also be implemented on the new algorithms and using the new kernels developed in this paper.

In all the examples, the computational domain is a discretization of $[0, 1]^d$, with $d = 2$ or 3.

3.5.1 Two-phase, anisotropic flows

(a) As an initial, modest test of the proposed kernel constructions, consider the task of simulating anisotropic, two-phase curvature flow in 2D given by (1.9), with the following

choice of surface tension and mobility:

$$\sigma(x_1, x_2) = \sqrt{x_1^2 + 4x_2^2} \text{ and } \mu(x_1, x_2) = 1. \tag{3.53}$$

Note that the corresponding Wulff shape is an ellipse in this case:

$$W_\sigma = \left\{ (x_1, x_2) \,:\, x_1^2 + \frac{1}{4} x_2^2 \leq 1 \right\}. \tag{3.54}$$

The construction of Section 3.4.1 yields the following positive convolution kernel: K given in polar coordinates by (3.24) with

$$\alpha(\theta) = \frac{m_2^{\frac{1}{2}}}{4m_0^{\frac{3}{2}}} \big(1 + 3\sin^2\theta\big)^{\frac{3}{4}} \text{ and } \beta(\theta) = \frac{m_2^{\frac{1}{2}}}{2m_0^{\frac{1}{2}}} \big(1 + 3\sin^2\theta\big)^{\frac{3}{4}}. \tag{3.55}$$

When η in the definition (3.24) of K is given by (3.22), the relevant moments (3.23) appearing in (3.55) are approximately

$$m_0 \approx 0.3403 \text{ and } m_2 \approx 0.3737. \tag{3.56}$$

Figure 3.1 shows the result of computation starting with a circle of radius $\frac{1}{4}$ as the initial condition, and ending at time $t = \frac{1}{80}$. The red curves are the result of threshold dynamics with the new kernels, with a spatial discretization of 256×256 and using 25 time steps of size $5 \cdot 10^{-4}$. The front tracking result, serving as our benchmark and shown in blue, used 128 points to discretize the curve and required 20480 time steps.

(b) A more interesting anisotropy for numerical exploration is

$$\sigma(x_1, x_2, x_3) = \max\big\{|x_1|, |x_2|, |x_3|\big\} \tag{3.57}$$

i.e. the ℓ^∞ norm that has as its Wulff shape the octahedron:

$$W_\sigma = \big\{(x_1, x_2, x_3) \,:\, |x_1| + |x_2| + |x_3| \leq 1\big\}. \tag{3.58}$$

Consider this with e.g. the constant mobility $\mu = 1$. Since W_σ is *not* a zonoid, according to Theorem 1.1.3, there exists no positive convolution kernel using which threshold dynamics scheme (1.14) & (1.15) can even approximate the corresponding flow. The new, fully general kernel construction of Section 3.5.1, however, easily yields a Schwartz class kernel K with positive Fourier transform that is consistent with this choice of anisotropy and mobility. Figure 3.2 shows evolution of a cube under a volume preserving version [76, 58]

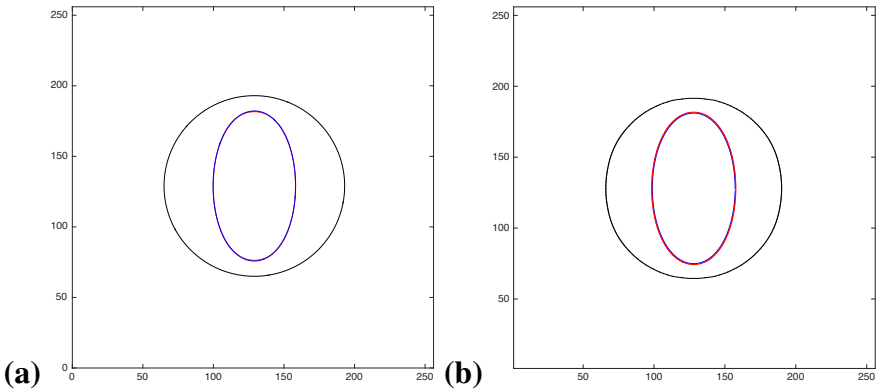

Figure 3.1: Evolution of an initial circle (black) under motion (1.9) with surface tension and mobility given by (3.53), computed using threshold dynamics algorithm, Algorithm 1.1 and the convolution kernels from Sections 3.4.2 and 3.4.1 (red), compared against benchmark result obtained with front tracking (blue). **(a)** Using the convolution kernel with positive Fourier transform of Section 3.4.2. **(b)** Using the positive convolution kernel of Section 3.4.1.

of Algorithm 1.1 as implemented in [78] or [25] using the kernel construction of Section 3.5.1 using this anisotropy and mobility pair.

3.5.2 Multi-phase, anisotropic flows

Consider the initial three phase configuration shown in Figure 3.4 (a), subject to the following surface tension and mobility pairs:

$$\begin{aligned}
\sigma_{1,2}(x_1,x_2) &= \sqrt{x_1^2+x_2^2} & \mu_{1,2}(x_1,x_2) &= 1,\\
\sigma_{1,3}(x_1,x_2) &= \sqrt{\frac{1}{4}x_1^2+x_2^2}+\sqrt{x_1^2+\frac{1}{4}x_2^2} & \mu_{1,3}(x_1,x_2) &= \frac{2x_1^2+3x_2^2}{4\sqrt{x_1^2+x_2^2}}\\
\sigma_{2,3}(x_1,x_2) &= \sqrt{x_1^2+\frac{25}{16}x_2^2} & \mu_{2,3}(x_1,x_2) &= 1.
\end{aligned}$$

The corresponding Wulff shapes for these anisotropies are shown in Figure 3.5.

The positive kernel construction given in Section 3.4.1 yields convolution kernels of the form

$$K_{i,j}(r,\theta) = \alpha_{i,j}(\theta)\eta\big(r\beta_{i,j}(\theta)\big)$$

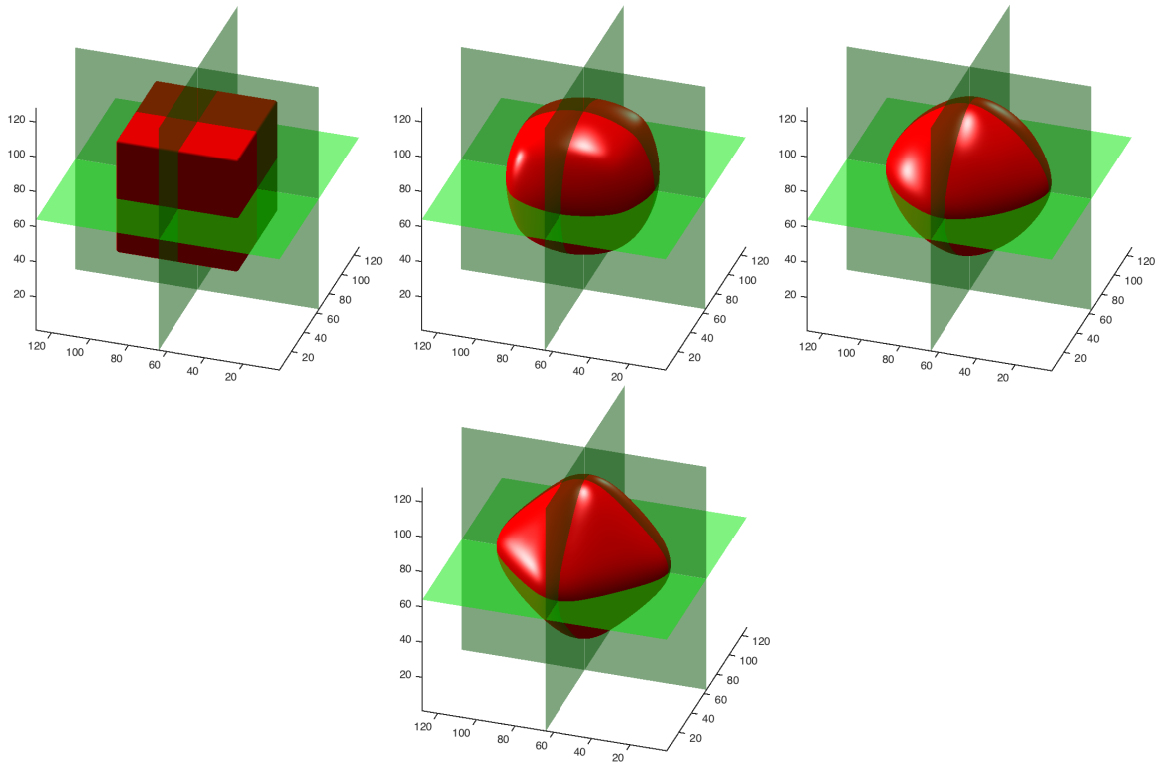

Figure 3.2: Evolution of a cube under volume preserving weighted mean curvature flow towards its Wulff shape the octahedron, with surface tension given by the ℓ^∞ norm and *constant* mobility. The corresponding kernel was obtained from the construction of Section 3.4.2. Compare with a similar experiment in [25] that uses a different convolution kernel that has the same surface tension but different mobility.

where, for example, $\alpha_{1,3}$ and $\beta_{1,3}$ are given by

$$\alpha_{1,3}(\theta) = \frac{m_2^{\frac{1}{2}}}{2m_0^{\frac{3}{2}}} \frac{\left(\cos^2(\theta) + \frac{1}{4}\sin^2(\theta)\right)^{\frac{3}{4}}\left(\frac{1}{4}\cos^2(\theta) + \sin^2(\theta)\right)^{\frac{3}{4}}}{\sqrt{\left(\cos^2(\theta) + \frac{1}{4}\sin^2(\theta)\right)^{\frac{3}{4}} + \left(\frac{1}{4}\cos^2(\theta) + \sin^2(\theta)\right)^{\frac{3}{4}}}} \times \left(\frac{1}{2} + \frac{1}{4}\sin^2(\theta)\right)^{-\frac{3}{2}}$$

and

$$\beta_{1,3}(\theta) = \frac{2m_2^{\frac{1}{2}}}{m_0^{\frac{1}{2}}} \frac{\left(\cos^2(\theta) + \frac{1}{4}\sin^2(\theta)\right)^{\frac{3}{4}}\left(\frac{1}{4}\cos^2(\theta) + \sin^2(\theta)\right)^{\frac{3}{4}}}{\sqrt{\left(\cos^2(\theta) + \frac{1}{4}\sin^2(\theta)\right)^{\frac{3}{4}} + \left(\frac{1}{4}\cos^2(\theta) + \sin^2(\theta)\right)^{\frac{3}{4}}}} \times \left(\frac{1}{2} + \frac{1}{4}\sin^2(\theta)\right)^{-\frac{1}{2}}$$

Figure 3.3 shows plots of these kernels.

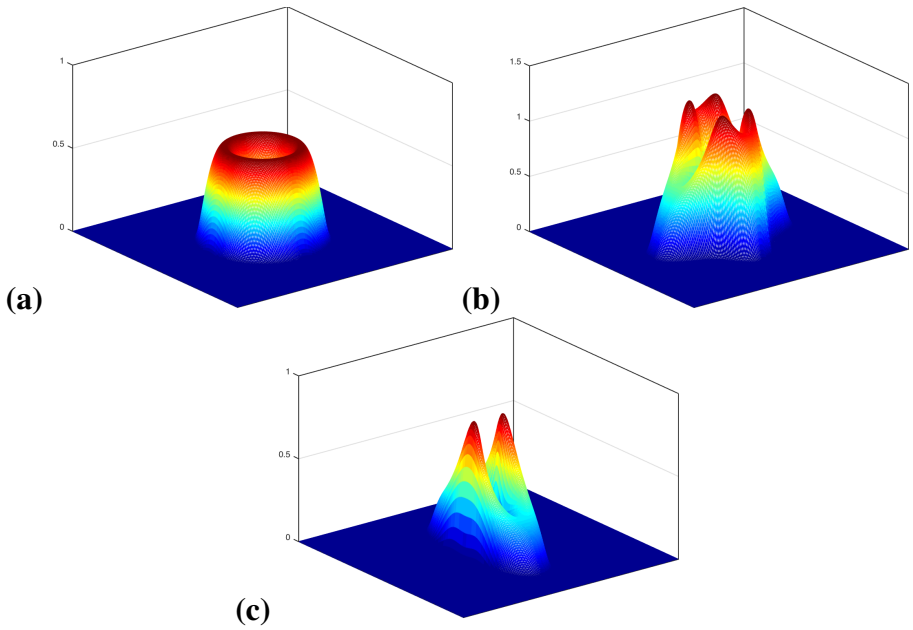

Figure 3.3: The kernels **(a)** $K_{1,2}$, **(b)** $K_{1,3}$, **(c)** and $K_{2,3}$ obtained from the construction of Section 3.4.1 for the surface tensions and mobilities used in the triple junction example of Figure 3.4.

Figure 3.4 (b) shows the resulting configuration at $t = 0.01$ computed using these kernels in threshold dynamics Algorithm 2.2 on a 256×256 grid with 20 time steps of size $\delta t = 5 \cdot 10^{-4}$, and compared to the benchmark front tracking result computed using 200 grid points along each one of the three curves and 8000 time steps of size $\delta t = 1.25 \cdot 10^{-6}$. There is very good agreement. We note that the kernels of Section 3.4.2 could have also been used in this example, since positivity of the convolution kernels *or* their Fourier transforms is sufficient for Algorithm 2.2 to dissipate the multiphase non-local energy (1.21). Although numerical experiments with these kernels suggest convergence to the correct evolution, the error appears to be dramatically larger than that of using kernels of Section 3.4.1. We leave finding a more accurate version of the construction in Section 3.4.2 to a future study, and recommend kernels of Section 3.4.1 over them in the multiphase setting instead.

Finally, Figure 3.6 demonstrates how the seamless behavior through topological changes of the original threshold dynamics algorithm of [62] is entirely preserved by its extensions to the fully anisotropic, multiphase setting provided by Algorithms 2.2 and 2.3. An initial condition consisting of 27 phases, obtained from a Voronoi construction for points chosen uniformly at random in a periodic domain of size $128 \times 128 \times 128$, is evolved via Algorithm 2.2 using two different sets of surface tensions and mobilities: one in which all mobilities and surface tensions are equal and isotropic (where algorithm of [62] can be used), and another in which there is one distinguished phase, $i = 1$, whose interfaces with other phases

are anisotropic both in mobility and surface tension. Myriad topological changes occur on the surface of the "grain", as quadruple points collide and split, and existing facets are destroyed and new ones are formed.

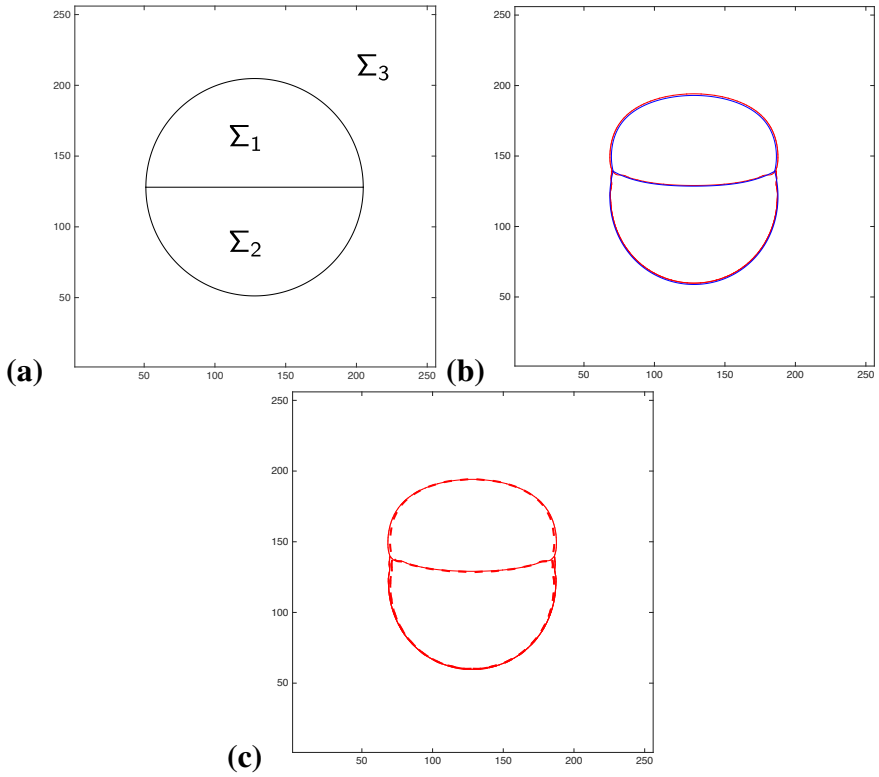

Figure 3.4: Evolution of a three-phase configuration with a pair of triple junctions under anisotropic curvature flow; each interface has a distinct prescribed surface tension (two of them anisotropic), two have constant mobility, and one has a normal dependent prescribed mobility, as described in detail in Section 3.5.2. **(a)** The initial condition. **(b)** Final configuration computed using threshold dynamics Algorithm 2.2 and the positive kernel construction presented in Section 3.4.1 (red), compared to the benchmark result computed using front tracking (blue). **(c)** The same evolution computed using two different threshold dynamics algorithms: Algorithm 2.2 shown in thin, solid vs. Algorithm 2.3 shown in thick, dashed line. Algorithm 2.2 is faster, but Algorithm 2.3 has guaranteed unconditional gradient stability for essentially *any* collection of N-choose-2 anisotropic surface tension and mobility pairs.

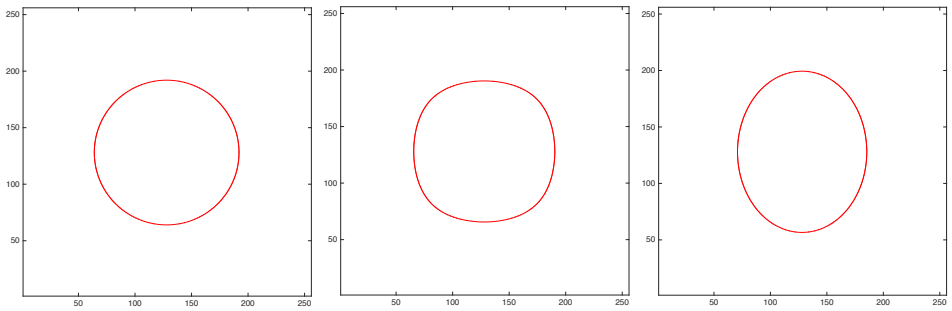

Figure 3.5: The Wulff shapes corresponding to the anisotropies used in the triple junction example. From left to right: $W_{\sigma_{1,2}}$, $W_{\sigma_{1,3}}$, and $W_{\sigma_{2,3}}$.

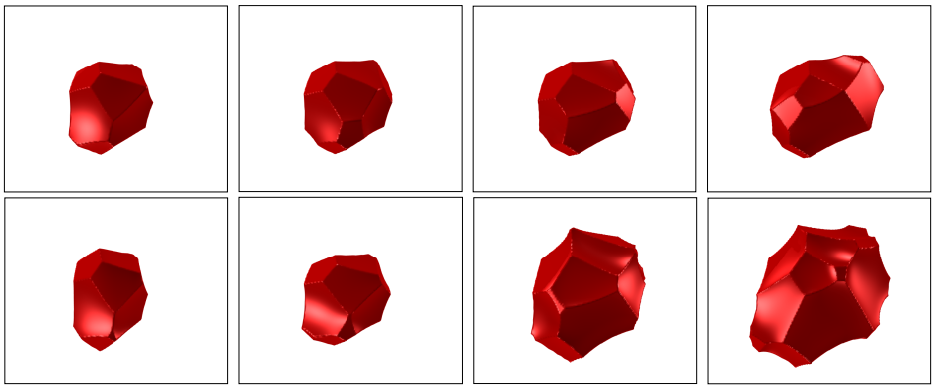

Figure 3.6: Evolution of phase $i = 1$ out of a total of 27 phases, from two different simulations starting from the same initial condition. Top row: $\sigma_{i,j}(\theta,\phi) = \mu_{i,j}(\theta,\phi) = 1$ for all $i \neq j$. Bottom row: $\sigma_{i,1}(\theta,\phi) = 1.1\sqrt{1+0.3\cos^2\phi}$ and $\mu_{i,1}(\theta,\phi) = \frac{1}{1.1}\sqrt{1+0.3\cos^2\phi}$ for $i \neq 1$, and $\sigma_{i,j}(\theta,\phi) = \mu_{i,j}(\theta,\phi) = 1$ for $i \neq 1$ and $j \notin \{i,1\}$. Many topological changes occur on the surface of the "grain", where quadruple points can be seen to collide and split numerous times.

3.6 Conclusions

We have presented two simple and practical methods to construct convolution kernels to be used in threshold dynamics schemes for curvature motion in two or three dimensions. Our constructions allow, for the first time, specifying a possibly anisotropic surface tension and possibly anisotropic mobility simultaneously: these are encoded into the convolution kernel. Combining the new kernel constructions with extensions of threshold dynamics algorithms presented in [26], we arrived at unconditionally gradient stable schemes for fully anisotropic, multiphase motion by weighted mean curvature of networks in two and three dimensions. This level of generality allows specifying n-choose-2 anisotropic surface tensions and n-choose-2 anisotropic mobilities for a network of n phases. Numerical ex-

periments indicate that in the multiphase setting, one of our new kernel constructions gives significantly more accurate results than the other.

Along the way, our work has also elucidated necessary and sufficient conditions for the positivity of the convolution kernel used in a threshold dynamics scheme in terms of the surface tension and mobility factors of the desired weighted mean curvature flow.

CHAPTER 4

Auction Dynamics: A Volume Preserving MBO Scheme [3]

4.1 Auction Dynamics

We now wish to extend the variational framework of the heat content energy to volume preserving MBO schemes. Suppose we have a partition $\mathbf{\Sigma}$ where each phase has some volume $m(\Sigma_\ell) = v_\ell$ with respect to the Lebesgue measure m on D. We now wish to evolve $\mathbf{\Sigma}$ by curvature flow while keeping the volume of each phase fixed. The natural approach is to solve (2.29) with the additional constraint that the volume of each phase must stay fixed. Thus the thresholding step is instead replaced with the following minimization problem

$$\arg\min_{\boldsymbol{u}\in\mathcal{K}_N} \sum_{\ell=1}^{N} \int_D \psi_\ell^k(x) u_\ell(x) dx \quad \text{s.t.} \quad \int_D u_\ell(x) dx = v_\ell. \tag{4.1}$$

If we incorporate the volume constraints with a Lagrange multiplier $\boldsymbol{\lambda}$ we see that the solution to (4.1) is a partition $\mathbf{\Sigma}$ given by a $\boldsymbol{\lambda}^*$ shifted thresholding

$$\Sigma_i = \{x \in D : i = \arg\min_{1\le\ell\le N} \psi_\ell(x) - \lambda_i^*\} \tag{4.2}$$

where $\boldsymbol{\lambda}^*$ is the optimal Lagrange multiplier. It then follows essentially immediately from [74] that the scheme is consistent with volume preserving weighted curvature flow. Furthermore, if $\hat{K} \geq 0$ and the surface tensions $\boldsymbol{\sigma}$ satisfy a triangle inequality then the scheme is also unconditionally stable [27].

Now, there is a difficulty that we need to address. If $N > 2$, finding the optimal Lagrange multiplier $\boldsymbol{\lambda}^*$ is not trivial (if $N = 2$ one can find the multiplier by simply sorting the convolution values). This task is particularly difficult if one insists on solving for the

[3] Joint work with Selim Esedoḡlu and Ekaterina Merkurjev [50].

Lagrange multiplier and the configuration $\mathbf{\Sigma}$ simultaneously (as we do). Our approach is to connect (4.1) to the assignment problem, a famous linear programming problem with efficient solutions. The assignment problem is typically posed as a maximization problem, thus as a first step we will replace (4.1) with the equivalent problem (4.3)

$$\arg\max_{\boldsymbol{u}\in\mathcal{K}_N} \sum_{\ell=1}^{N} \int_D a_\ell(x)u_\ell(x)dx \quad \text{s.t.} \quad \int_D u_\ell(x)dx = v_\ell, \tag{4.3}$$

where $a_\ell(x) = (1 - \psi_\ell^k(x))$. However, rather than working with problem (4.3) directly, we will consider a discretized version. Discretization is natural, as any implementation of the scheme must be carried out on a finite grid. Discretization also allows us to more clearly connect our approach to the assignment problem, which is typically posed over a finite dimensional vector space. Let $D_n = \{x_1, \ldots, x_n\} \subset D$ be some n point discretization of D. We discretize the volume constraints by requiring each phase to occupy V_ℓ points, where V_ℓ are integers chosen so that $\sum_{\ell=1}^N V_\ell = n$ and the ratios $V_\ell/n \approx v_\ell/m(D)$ are as close as possible. Since the convolution values $a_\ell(x) = (1 - \psi_\ell^k(x))$ are smooth functions, they have a well defined restriction to D_n. Finally, the discrete analogue of $\mathcal{K}_N$ is the set of functions $\{\boldsymbol{u} : D_n \to [0,1]^N : \sum_{\ell=1}^N u_\ell(x) = 1\}$, which may also be represented as $\{\boldsymbol{u} \in [0,\infty)^{n\times N} : \sum_{\ell=1}^N u_\ell(x) = 1\}$. Using the latter representation we arrive at

$$\arg\max_{\mathbf{u}\geq\mathbf{0}} \sum_{\ell=1}^{N} \sum_{x\in D_n} a_\ell u_\ell(x) \quad \text{s.t.} \sum_{x\in D_n} u_\ell(x) = V_\ell, \sum_{\ell=1}^{N} u_\ell(x) = 1. \tag{4.4}$$

In this form, problem (4.4) can be viewed as a special case of a family of linear programming problems. This family of problems stems from the minimum cost flow problem, and includes famous problems such as the assignment problem, the transportation problem and the maximum flow problem. We choose to focus on the assignment problem, as it is the simplest of the problems and can be solved with an intuitive economic approach.

4.1.1 The assignment problem

Given two disjoint sets X and I of equal size r and a weight function $w : X \times I \to \mathbb{R}$, the assignment problem seeks to find a one to one matching $M = \{(x_1, i_1), \ldots, (x_r, i_r)\}$ of X and I (i.e. a bijection), such that the total weight of the matching

$$\sum_{(x,i)\in M} w_i(x) \tag{4.5}$$

is maximized. By representing the matching as a binary vector $\mathbf{z}$ where $z_i(x) = 1$ if (x, i) are matched and $z_i(x) = 0$ otherwise, we can restate the assignment problem as the following optimization problem

$$\max_{\mathbf{z}:X\times I\to\{0,1\}} \sum_{x\in X}\sum_{i\in I} w_i(x)z_i(x) \quad \text{s.t. } \sum_{x\in X} z_i(x) = 1, \sum_{i\in I} z_i(x) = 1. \tag{4.6}$$

If we relax the binary constraint on $\boldsymbol{z}$ then (4.6) becomes the following linear programming problem

$$\max_{\boldsymbol{z}\geq\mathbf{0}} \sum_{x\in X}\sum_{i\in I} w_i(x)z_i(x) \quad \text{s.t. } \sum_{x\in X} z_i(x) = 1, \sum_{i\in I} z_i(x) = 1. \tag{4.7}$$

It turns out that the relaxation is exact, and we may substitute (4.7) for (4.6). This follows from the fact that the solution to a bounded and feasible linear programming problem always includes a vertex of the feasible polytope. The relaxed linear constraint set is the polytope $\{\boldsymbol{z} \geq \mathbf{0} : \sum_{x\in X} z_i(x) = 1, \sum_{i\in I} z_i(x) = 1\}$. The vertices of the polytope are precisely the vectors $\boldsymbol{z}$ whose entries are binary.

Now observe that problem (4.4) is a special case of (4.7), with respect to a particular choice of weights $w_i(x)$. We can obtain (4.4) from a generic instance of (4.7) by letting $X = D_n$ splitting I into N similarity classes $\{S_\ell\}_{\ell=1}^N$ each of size V_ℓ and setting $w_i(x) = a_\ell(x)$ for every $i \in S_\ell$. With those choices (4.7) becomes

$$\max_{\boldsymbol{z}\geq\mathbf{0}} \sum_{x\in D_n}\sum_{\ell=1}^{N} a_\ell(x)\sum_{i\in S_\ell} z_i(x) \quad \text{s.t. } \sum_{x\in D_n} z_i(x) = 1, \sum_{i\in I} z_i(x) = 1. \tag{4.8}$$

Now if we define $u_\ell(x) = \sum_{i\in S_\ell} z_i(x)$, we can relax the constraints to $\sum_{x\in D_n} u_\ell(x) = V_\ell$ and $\sum_{\ell=1}^N u_\ell(x) = 1$ without changing the value of the problem. Thus (4.4) can be obtained as a special case of (4.7). In fact, the two problems are actually equivalent, a generic instance of (4.4) can also be transformed into a specific instance of (4.7).

For the remainder of this chapter we will focus on our special case (4.4) of the assignment problem (see [10] for a similar discussion of the classic formulation (4.7)). We will interchangeably represent matchings as vectors $\boldsymbol{u}$ in the feasible polytope and as partitions $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_N)$ of D_n. Our goal for the remainder of this subsection is to develop an intuition for (4.4), and develop the necessary setup for the auction algorithm in Section 4.1.2.

It is particularly instructive to give a practical interpretation of (4.4). Imagine that each phase is an institution that offers a limited number of memberships. For example, the phases may be gyms, or clubs, or different Costco locations, etc. Imagine that the points $x \in D_n$ are people, and each person would like to become a member of some

phase. No person wants to have a membership in more than one phase, and each phase only has V_ℓ memberships available. Finally, imagine that the coefficients $a_\ell(x)$ represent how much person x wants to be a member of phase ℓ. Now we can think of the solution to the assignment problem as the matching of people and phases that maximizes the total satisfaction of the population. Ideally, each person would like to become a member of their favorite phase. However, this is not possible if more than V_ℓ people want to be members of some phase ℓ. The main difficulty of the assignment problem is in understanding how to correctly handle these conflicts.

An interesting approach is to attempt to assign the memberships according to a market mechanism. Imagine that each phase ℓ has a membership price p_ℓ, and if person x is a member of ℓ then they must pay p_ℓ. This can help to resolve conflicts by making the most popular phases more expensive. Assuming that every person acts in their own best interest, x will want to buy a membership at the phase offering the best value, i.e. x wants to be a member of any phase

$$\ell^* \in \ell_{cs}(x, \boldsymbol{p}) = \arg\max_{1 \le \ell \le N} a_\ell(x) - p_\ell. \tag{4.9}$$

We are now led to a very interesting question: does there exist an equilibrium price vector $\boldsymbol{p}_*$ such that assigning memberships according to (4.9) gives a feasible matching? The answer to this question is yes, and better yet the resulting assignment is optimal.

The connection between the assignment problem and the equilibrium price vector $\boldsymbol{p}_*$ comes from the duality theory of linear programming. As it turns out, the equilibrium price $\boldsymbol{p}_*$ is in fact the optimal solution to the dual of the assignment problem. In addition to the prices $\boldsymbol{p}$, the dual problem introduces a set of variables $\pi(x)$ for each $x \in D_n$. The dual problem is

$$\min_{\boldsymbol{p} \in \mathbb{R}^N, \pi \in \mathbb{R}^n} \sum_{\ell=1}^{N} p_i + \sum_{x \in D_n} \pi(x) \quad \text{s.t. } \pi(x) + p_\ell \ge a_\ell(x). \tag{4.10}$$

Note that the optimal value of $\boldsymbol{\pi}$ is entirely determined by $\boldsymbol{p}$. Given any $\boldsymbol{p}$ the best choice for $\boldsymbol{\pi}$ is to set $\pi(x) = \max_{1 \le \ell \le N} a_\ell(x) - p_\ell$. This shows that $\pi(x)$ is exactly the value of the best deal offered to x by any phase.

Our earlier statements about the equilibrium price vector $\boldsymbol{p}_*$ can be justified by invoking the complementary slackness (CS) condition. According to CS, a feasible assignment $\boldsymbol{u}$ and a feasible dual pair $(\boldsymbol{p}, \boldsymbol{\pi})$ are optimal for their respective problems if and only if

$$\sum_{x \in D_n} \sum_{\ell=1}^{N} u_\ell(x)(p_\ell + \pi(x) - a_\ell(x)) = 0. \tag{4.11}$$

Recalling the best choice for $\boldsymbol{\pi}$, (4.11) implies that in the optimal matching every person

is assigned a membership which satisfies the market strategy (4.9) using the optimal price vector $\boldsymbol{p}_c$. This implies that the equilibrium price $\boldsymbol{p}_*$ exists and $\boldsymbol{p}_* = \boldsymbol{p}_c$.

Now suppose that we have some price vector $\boldsymbol{p}$ which is not optimal for the dual problem. By CS, it will not be possible to assign every membership according to (4.9), there will necessarily be conflicts. However, we can attempt to construct a partial assignment (partial matching). A partial assignment matches a subset of people in $S \subset D_n$ to phases $\{1, \ldots, N\}$ while ensuring that no more than V_ℓ people are assigned to any phase. A partial matching can be represented as a partition $\boldsymbol{\Sigma}$ of S into N phases $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_N)$ such that $|\Sigma_\ell| \leq V_\ell$ for every phase ℓ. Given a partial matching $\boldsymbol{\Sigma}$, if $x \in \Sigma_\ell$ then it will be notationally convenient to say that the pair (x, ℓ) is in the matching. A partial assignment $\boldsymbol{\Sigma}$ and a price $\boldsymbol{p}$ satisfies CS if for every phase ℓ and every member $x \in \Sigma_\ell$ the pair (x, ℓ) satisfies (4.9).

The most efficient algorithms for the assignment problem have the same basic structure. They generate a sequence of price vectors $\boldsymbol{p}^k$ and partial matchings $\boldsymbol{\Sigma}^k$ such that $\boldsymbol{\Sigma}^k$ and $\boldsymbol{p}^k$ satisfy CS. Each stage of the algorithm either increases the size of the partial matching (a larger subset of D_n matched) or improves the prices (with respect to the dual problem value). Since CS is preserved at every step, if the partial matching becomes a complete matching then it is an optimal solution to the assignment problem.

We will solve the assignment problem using auction algorithms. Auction algorithms have a simple intuitive structure, are easy to code, and have excellent performance. The main advantage of auction algorithms over the well-known Hungarian algorithm [53, 68]is that auction algorithms perform local modifications of $\boldsymbol{\Sigma}^k$ and $\boldsymbol{p}^k$ at every step, whereas the Hungarian algorithm may need to consider global modifications.

4.1.2 Auction algorithms

In [9] Bertsekas developed the auction algorithm for solving the classic assignment problem (4.7). Since the original paper, Bertsekas and collaborators have improved upon the computational aspects of the auction algorithm, and extended it to more general minimum cost flow problems (see [11] or [10] for an exhaustive reference on auction algorithms). The most important references for this work are [12] and [13]. In [12] Bertsekas and Castanon modified the auction algorithm to more efficiently handle assignment problems with multiple identical objects as in (4.4). In [13], Bertsekas, Castanon, and Tsaknakis introduced the reverse auction algorithm for asymmetric assignment problems, which we will use in Section 4.1.3.

The basic idea of the auction algorithm is to drive price modifications and augmenta-

tions of the partial matching by simulating an auction. In order to obtain a membership, each person x must submit a bid $b(x)$ to the phase of their choice. At the start of the auction, the price of a membership at each phase ℓ is set to a starting value p_ℓ^0 according to some initial price vector $\boldsymbol{p}^0$. As in a real life auction, if a person x submits a bid $b(x)$ to a phase ℓ the bid must exceed or match the current price p_ℓ. Using the CS condition (4.9), we can split the phases into three sets, the high demand phases H, the low demand phases L, and the equilibrium phases E. The high demand phases $\ell \in H$ have more than V_ℓ people who would like to purchase a membership, the low demand phases $\ell \in L$ have fewer than V_ℓ people and the equilibrium phases $\ell \in E$ have exactly V_ℓ people. Everyone who wants a low demand or equilibrium membership can submit a bid and immediately be accepted into the initial partial matching, but there is a conflict at the high demand phases. The conflict is resolved by choosing the people who have submitted the largest bids. At any step of the algorithm, if ℓ is a high demand phase then the set Σ_ℓ consists of the V_ℓ people who have submitted the largest bids for phase ℓ. As people submit bids, the prices of the high demand phases will rise. Eventually, this will incentivize unmatched people to switch their bid to a cheaper phase that may offer a better bang for their buck. The algorithm terminates once all of the phases are in equilibrium.

Now we need to discuss pricing and bidding strategies. Each phase is restricted to setting one uniform membership price, regardless of how large individual bids may be. Assuming that a phase does not want to lose members, the price should be set to the amount that the least committed member is willing to pay. This amount is the lowest bid that a phase received thus far. To make this strategy consistent across all phases, assume that the empty spots in every low demand phase $\ell \in L$ are filled by phantom members who all bid the starting price p_ℓ^0.

Finding a bidding strategy that guarantees termination of the algorithm and produces a complete matching satisfying CS turns out to be nontrivial. For a given price vector $\boldsymbol{p}$, if person x is a member of phase ℓ then we must have $\ell \in \ell_{cs}(x, \boldsymbol{p})$ to satisfy CS. This suggests that in the course of the auction, an unmatched person x should only submit bids to phases in $\ell_{cs}(x, \boldsymbol{p})$. The subtlety lies in the question, 'How much should person x be willing to bid?' Obviously, x does not want to overbid, otherwise prices may rise and a different phase will become optimal according to CS. The largest bid, $b(x)$, that x can submit to $\ell^* \in \ell_{cs}(x, \boldsymbol{p})$ while being guaranteed not to violate CS is

$$b(x) = p_{\ell^*} + (a_{\ell^*}(x) - p_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - p_{\ell_{\text{next}}}) \tag{4.12}$$

where

$$\ell_{\text{next}} \in \arg\max_{j \neq \ell^*} a_\ell(x) - p_\ell \tag{4.13}$$

is x's second most desirable phase. With this bid, x is willing to allow the price of ℓ^* to increase to at most the gap in value between the best and second best choice. While x is matched to ℓ^*, the price p_{ℓ^*} cannot increase beyond $b(x)$. Other prices are non-decreasing, thus for the duration that (x, ℓ^*) is part of the partial matching, this pair satisfies CS.

Unfortunately, this bidding strategy does not work. A problem occurs when there are multiple optimal objects for x, i.e. when $|\ell_{cs}(x)| > 1$. If this happens then both $\ell^*, \ell_{\text{next}} \in \ell_{cs}(x, \boldsymbol{p})$ and thus the gap $(a_{\ell^*}(x) - p_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - p_{\ell_{\text{next}}}) = 0$. In this case x is unable to raise the price of ℓ^*. This situation may lead to a price war. In a price war, multiple people compete for the same memberships without ever raising the prices, trapping the algorithm in an infinite loop.

To circumvent this difficulty, one must relax the complementary slackness condition. For a given price vector $\mathbf{p}$ and a small $\varepsilon > 0$ a matched pair (x, ℓ) satisfies the ε-complementary slackness condition (ε-CS) if

$$a_\ell(x) - p_\ell + \varepsilon \geq \max_{1 \leq j \leq N} a_\ell(x) - p_\ell. \tag{4.14}$$

It is now possible to create a bidding strategy that preserves ε-CS and guarantees that the algorithm will always terminate. As before, an unmatched x will only submit bids to $\ell^* \in \ell_{cs}(x, \boldsymbol{p})$, however now x can bid up to

$$b(x) = p_{\ell^*} + \varepsilon + (a_{\ell^*}(x) - p_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - p_{\ell_{\text{next}}}) \tag{4.15}$$

without overpaying according to ε-CS. Since $(a_{\ell^*}(x) - p_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - p_{\ell_{\text{next}}}) \geq 0$ this ensures that p_{ℓ^*} increases by at least ε. This mimics real life auctions where any bid must be larger than the current price by at least some fixed amount. Now starting from any initial price vector $\boldsymbol{p}_0$ the algorithm will be guaranteed to eventually terminate [9]. We now give our version of the auction algorithm, which is equivalent to the "similar object" auction

variant in [12].

Algorithm 4.1: Membership Auction [12]

Input: $\varepsilon > 0$, volumes $\boldsymbol{V}$, coefficients $\boldsymbol{a}$, initial prices $\boldsymbol{p}^0$ and people $x \in D_n$

Result: Final prices and complete ε-CS matching $(\boldsymbol{\Sigma}, \boldsymbol{p})$.

Initialization: For every $\ell \in \{1, \ldots, N\}$ mark all x as unassigned, set $\boldsymbol{p} = \boldsymbol{p}^0$, set $\boldsymbol{\Sigma} = \varnothing$;

while *some x is marked as unassigned* **do**

  for *each unassigned $x \in D_n$* **do**

    Calculate $\ell_{cs}(x, \boldsymbol{p})$ and choose some $\ell^* \in \ell_{cs}(x, \boldsymbol{p})$;

    Set $b(x) = p_{\ell^*} + \varepsilon + (a_{\ell^*}(x) - p_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - p_{\ell_{\text{next}}})$;

    if $|\Sigma_{\ell^*}| = V_{\ell^*}$ **then**

      Find $y = \arg\min_{z \in \Sigma_{\ell^*}} b(z)$;

      Remove y from Σ_{ℓ^*} and add x to Σ_{ℓ^*};

      Mark y as unassigned and mark x as assigned;

      Set $p_{\ell^*} = \min_{z \in \Sigma_{\ell^*}} b(z)$;

    else

      Mark x as assigned and add x to Σ_{ℓ^*};

      if $|\Sigma_{\ell^*}| = V_\ell$ **then**

        Set $p_{\ell^*} = \min_{z \in \Sigma_{\ell^*}} b(z)$;

      end

    end

  end

end

return $(\boldsymbol{\Sigma}, \boldsymbol{p})$

The output of the auction algorithm is a complete matching $\boldsymbol{\Sigma}$ satisfying ε-CS, and the final auction prices $\boldsymbol{p}$. Representing the matching as a binary vector $\boldsymbol{u}$ and using ε-CS we may conclude

$$\sum_{\ell=1}^{N} p_\ell + \sum_{x \in D_n} \max_{1 \le \ell \le N} [a_\ell(x) - p_\ell] \le n\varepsilon + \sum_{x \in D_n} \sum_{\ell=1}^{N} u_\ell(x) a_\ell(x). \tag{4.16}$$

Thus, by weak duality, the final assignment $\boldsymbol{u}$ is at most $n\varepsilon$ away from being optimal. In the special case where the coefficients $a_\ell(x)$ integers, an ε-CS matching is actually optimal for any $\varepsilon < \frac{1}{N}$ [12].

We now give a quick sketch of the complexity. Assume that at the start of the auction the prices were initialized to zero and the partial matching was empty. Let $C =$

$\max_{\ell \in \{1,\ldots,N\}, x \in D_n} a_\ell(x)$ be the largest coefficient. Suppose in the course of the algorithm that the price of some phase ℓ exceeds C. If the algorithm has not yet terminated then there must be some low demand phase with price zero. This implies that in the remainder of the auction no person x will ever bid on phase ℓ again, since there must be a phase offering a better value. Thus we have an upper bound on the price of any phase. Suppose that some phase ℓ is currently priced at p_ℓ, and consider the number of bids required to raise the price. The worst possible case occurs when every currently matched member has bid exactly p_ℓ (such a situation is highly degenerate and rarely appears in practical applications). In this case it will take exactly V_ℓ bids to raise the price. The price must rise by at least ε, thus, we can conclude that the algorithm will terminate after at most $NV\lceil C/\varepsilon \rceil$ bids where $V = \max_{1 \le \ell \le N} V_\ell$.

A straightforward implementation of the bidding steps in Algorithm 4.1 requires $O(V + N)$ operations. This can be sped up with special data structures. If we implement a priority queue for each Σ_ℓ we can complete a bid in $O(\log(V) + N)$ operations. In all of our applications V is several orders of magnitude larger than N, thus this gives considerable savings. Combining this with the estimate for the maximum number of bids we can conclude the algorithm has complexity $O\big(NV(\log(V) + N)C/\varepsilon\big)$. Note that due to the presence of the constant C this complexity is pseudo-polynomial rather than polynomial.

The complexity can be improved using the idea of epsilon scaling (noted in [9] and analyzed in [36, 37, 38]). Suppose that $(\boldsymbol{\Sigma}', \boldsymbol{p}')$ is a matching and a price vector satisfying $r\varepsilon$-CS for some $r > 1$. What happens if we use $\boldsymbol{p}'$ as the initial price vector when we run the auction algorithm with ε? Since any starting price is admissible, the algorithm will still produce a matching and price $(\boldsymbol{\Sigma}, \boldsymbol{p})$ satisfying ε-CS. However, if r is not too large then we should expect that $\boldsymbol{p}'$ and $\boldsymbol{p}$ are not too different. This suggests that the auction will not need to modify the prices very much, and thus the algorithm will terminate quickly. Epsilon scaling takes advantage of this idea by running the auction multiple times with successively smaller values of epsilon. The final price vector of the previous run is used as the initial price vector in the next run. Typically, one takes the sequence of decreasing epsilon values to be $\varepsilon_k = C/\alpha^k$ for some integer $\alpha > 1$, stopping once $\varepsilon_k < \frac{\delta}{n}$ for some small δ. Using ε scaling the complexity can be improved to a weakly polynomial bound. We refer our readers to [11] for the exact details and bounds using ε-scaling. For the problems that we consider, the complexity of the auction algorithm using ε-scaling appears to grow like $O\big(NV(\log(V) + N)\log(nC/\delta)\big)$ (see [10] or [11] for a heuristic explanation of this behavior).

Now we are ready to give the auction dynamics algorithm, Algorithm 4.2. Recall that our goal is to simulate the evolution of a configuration $\boldsymbol{\Sigma}$ under volume preserving curva-

ture flow for some time $t = m(\delta t)$. As we saw in the beginning of Section 4.1 we obtain a consistent and unconditionally stable scheme by solving the iteration

$$\mathbf{\Sigma}^{k+1} = \arg\min_{\Sigma} \mathcal{L}_{\delta t}(\mathbf{\Sigma}^k, \mathbf{\Sigma}) \quad \text{s.t.} \quad |\Sigma_\ell| = V_\ell \quad \text{for } 1 \leq \ell \leq N \tag{4.17}$$

m times. This amounts to repeatedly taking convolutions of the configuration $\mathbf{\Sigma}^k$ with a kernel K, and solving the assignment problem. As we have seen above, we can solve the assignment problem efficiently using auctions. Auction dynamics uses Algorithm 4.1 along with ε-scaling to quickly and accurately obtain a solution. We give the algorithm below.

Algorithm 4.2: Auction Dynamics

Input: Discrete domain D_n, initial configuration $\mathbf{\Sigma}$, surface tensions $\boldsymbol{\sigma}$, convolution kernel K, volumes $\boldsymbol{V}$, time step δt, number of steps m, auction error tolerance ε_{min}, epsilon scaling factor α, initial epsilon value ε_0.

Result: Final configuration $\mathbf{\Sigma}^m$

Initialization: Set $\mathbf{\Sigma}^0 := \mathbf{\Sigma}$, set $\bar{\varepsilon} = \varepsilon_{min}/n$;

for k *from* 0 *to* $m-1$ **do**

  Calculate the convolutions: $\psi_\ell^{k+1}(x) = \sum_{j \neq \ell} \sigma_{\ell j}(K_{\delta t} * \Sigma_j^k)(x)$;

  Calculate the assignment problem coefficients: $\boldsymbol{a}^{k+1} = 1 - \boldsymbol{\psi}^{k+1}$;

  Initialize prices $\boldsymbol{p} = \mathbf{0}$, and $\varepsilon = \varepsilon_0$;

  while $\varepsilon \geq \bar{\varepsilon}$ **do**

    Run Algorithm 4.1 (Membership Auction):

    $(\mathbf{\Sigma}_{\text{out}}, \boldsymbol{p}_{\text{out}}) = \text{Membership Auction}(\varepsilon, \boldsymbol{V}, \boldsymbol{a}^{k+1}, \boldsymbol{p}, D_n)$;

    Set $\boldsymbol{p} = \boldsymbol{p}_{\text{out}}$;

    Divide ε by α;

    if $\varepsilon < \bar{\varepsilon}$ **then**

      Set $\mathbf{\Sigma}^{k+1} = \mathbf{\Sigma}_{\text{out}}$;

    end

  end

end

return $\mathbf{\Sigma}^m$

4.1.3 Upper and lower volume bounds

In addition to strict volume preserving curvature flow, auction dynamics can be modified to allow the volume of each phase to fluctuate between some bounds. This will be particularly useful in our applications to machine learning.

Suppose that each phase ℓ must have at least B_ℓ members and at most U_ℓ members for

some integers B_ℓ and U_ℓ. To ensure that the resulting problem is feasible we will require $B_\ell \leq U_\ell$ and $\sum_{\ell=1}^N B_\ell \leq n \leq \sum_{\ell=1}^N U_\ell$. We will then need to solve the following modified version of the assignment problem

$$\max_{\mathbf{u}\geq \mathbf{0}} \sum_{\ell=1}^N \sum_{x\in D_n} a_\ell(x) u_\ell(x) \quad \text{s.t. } \sum_{\ell=1}^N u_\ell(x) = 1,\ B_\ell \leq \sum_{x\in D_n} u_\ell(x) \leq U_\ell. \tag{4.18}$$

This version of the problem introduces some complexities that were not present in (4.4) and will require a more sophisticated approach.

Previously, we examined and solved the assignment problem from the perspective of the people $x \in D_n$. The limited supply of memberships resulted in competition between the people, which we resolved by introducing prices and simulating an auction. The upper bounds fit nicely into this perspective. The upper bounds indicate that each phase has a limited number of memberships, however it is now possible that the total supply of memberships $\sum_{\ell=1}^N U_\ell$ exceeds the number of people n. The upper bounds will still induce competition between the people, but the oversupply of memberships means that the set of equilibrium prices will be larger. This will add a wrinkle of difficulty, as not all equilibrium prices will be dual optimal.

The lower bounds are fundamentally different and require a new perspective. Indeed, if some person x sees that there is an available membership in their most desirable phase ℓ, they will immediately join ℓ without caring if some other phase j is deficient (i.e. $|\Sigma_j| < B_j$). Instead, we must think about the lower bounds from the perspective of the phases. Imagine that each phase must sell B_ℓ memberships or they will go out of business. If a phase ℓ is having trouble attracting a sufficient number of people, it will have to introduce an incentive t_ℓ to entice people to join. As a result, the lower bounds induce a competition among the phases. Deficient phases will be forced to offer competing incentives to attract the necessary number of members. Thus, in order to satisfy the lower bounds, we will need to run a reverse auction [13] where the phases bid on the people.

To properly understand the interaction between the prices and incentives, we introduce the dual problem

$$\min_{\boldsymbol{p}\geq \mathbf{0}, \boldsymbol{t}\geq \mathbf{0}, \boldsymbol{\pi}\in\mathbb{R}^n} \sum_{\ell=1}^N p_\ell U_\ell - t_\ell B_\ell + \sum_{x\in D_n} \pi(x) \quad \text{s.t.} \quad p_\ell - t_\ell + \pi(x) \geq a_\ell(x). \tag{4.19}$$

As before, we will use the interplay between the primal and dual problems to drive the search for the optimal solution. The key of course will be the complementary slackness condition. The complementary slackness condition for (4.18) and (4.19) states that an

assignment $\boldsymbol{u}$ and dual variables $(\boldsymbol{p}, \boldsymbol{t}, \boldsymbol{\pi})$ are optimal for their respective problems if and only if

$$\begin{aligned}&\sum_{\ell=1}^{N} \sum_{x \in D_n} u_\ell(x)(a_\ell(x) - p_\ell + t_\ell - \pi(x)) \\ &+ \sum_{\ell=1}^{N} p_\ell(U_\ell - \sum_{x \in D_n} u_\ell(x)) + \sum_{\ell=1}^{N} t_\ell(\sum_{x \in D_n} u_\ell(x) - B_\ell) = 0.\end{aligned} \tag{4.20}$$

Recall that $\boldsymbol{\pi}$ is determined by $\boldsymbol{p}$ and $\boldsymbol{t}$ and is given by $\pi(x) = \max_{1 \le \ell \le N} a_\ell(x) + t_\ell - p_\ell$. Now we can recognize that the complementary slackness condition has a simple intuitive explanation. The first sum states that each person should be assigned to the optimal phase based on prices and incentives (this should feel familiar). The second sum states that phases charging membership prices must have the maximum number of members U_ℓ (i.e. no overpriced phases). Similarly, the third sum states that the phases offering incentives must have the minimal number of members B_ℓ (i.e. no over-incentivized phases).

To ensure our auctions do not stall, we will once again turn to the ε-CS condition. For this problem, we will say that a partial matching $\boldsymbol{\Sigma}$ and a price-incentive pair $(\boldsymbol{p}, \boldsymbol{t})$ satisfy ε-CS if every matched pair (x, ℓ) satisfies

$$a_\ell(x) - p_\ell + t_\ell + \varepsilon \ge \max_{1 \le j \le N} a_j(x) - p_j + t_j. \tag{4.21}$$

As before, we can recognize this ε-CS condition as an ε relaxed version of the first sum in (4.20). Unfortunately, the other two terms in (4.20) do not have useful ε relaxations. As a result, we will need to carefully ensure that our auctions will satisfy the other two terms exactly. We will say that a price $\boldsymbol{p}$ (an incentive $\boldsymbol{t}$) is admissible for a matching $\boldsymbol{\Sigma}$ if the second (third) term of (4.20) is satisfied.

We will solve (4.18) in two stages. First we will run Algorithm 4.3, a forward auction algorithm similar to Algorithm 4.1, where the people compete for memberships. This will produce a complete ε-CS matching satisfying the upper bound constraints but possibly violating the lower bound constraints (we will call this upper feasible). Algorithm 4.3 differs from Algorithm 4.1, as it simultaneously runs a mechanism to ensure that no phase is over-incentivized. In the second stage we will feed the result of the first stage into a reverse auction, Algorithm 4.4, where the phases compete for people. This will produce an ε-CS matching that is both upper and lower feasible. In addition, Algorithm 4.4 will have a mechanism to prevent phases from becoming overpriced. As a result, the final output will be a complete and feasible ε-CS matching $\boldsymbol{\Sigma}$ with admissible prices and incentives $(\boldsymbol{p}, \boldsymbol{t})$. This will be enough to conclude that $\boldsymbol{\Sigma}$ solves (4.18) with error at most $n\varepsilon$. In the

special case that the coefficients $\boldsymbol{a}$ are all integers, the argument used in [12] can be easily generalized to show that the solution is optimal if $\varepsilon < \frac{1}{N}$.

Algorithm 4.3 is a relatively straightforward adaptation of the similar object auctions and the asymmetric assignment auctions found in [11]. On the other hand, Algorithm 4.4 appears to have a different structure than the reverse auctions considered in [11]. Indeed, in our reverse auction we choose to work with prices and incentives rather than the profit variable $\boldsymbol{\pi}$. We find that working with prices and incentives leads to a much faster runtime when $N << n$. Since both algorithms are highly specialized for our current problem, we provide proofs that they terminate and have the desired properties.

Algorithm 4.3: Upper Bound Auction

Input: $\varepsilon > 0$, bounds $\boldsymbol{B}, \boldsymbol{U}$, coefficients $\boldsymbol{a}$, initial prices $\boldsymbol{p}^0$, initial incentives $\boldsymbol{t}^0$ and people $x \in D_n$

Result: Prices $\boldsymbol{p}$, admissible incentives $\boldsymbol{t}$, and complete ε-CS matching $\boldsymbol{\Sigma}$ satisfying upper bounds.

Initialization: Mark all x as unassigned, set $\boldsymbol{d} = \boldsymbol{p}^0 - \boldsymbol{t}^0$, set $\boldsymbol{\Sigma} = \varnothing$;

while *some x is marked as unassigned* **do**

  for *each unassigned $x \in D_n$* **do**

    Calculate $\ell_{cs}(x, \boldsymbol{p})$ and choose some $\ell^* \in \ell_{cs}(x, \boldsymbol{d})$;

    Set $b(x) = d_{\ell^*} + \varepsilon + (a_{\ell^*}(x) - d_{\ell^*}) - (a_{\ell_{\text{next}}}(x) - d_{\ell_{\text{next}}})$;

    if $|\Sigma_{\ell^*}| = U_{\ell^*}$ **then**

      Find $y = \arg\min_{z \in \Sigma_{\ell^*}} b(z)$;

      Remove y from Σ_{ℓ^*} and add x to Σ_{ℓ^*};

      Mark y as unassigned and mark x as assigned;

      Set $d_{\ell^*} = \min_{z \in \Sigma_{\ell^*}} b(z)$;

    else if $|\Sigma_\ell| = B_\ell$ *and* $d_\ell < 0$ **then**

      Find $y = \arg\min_{z \in \Sigma_{\ell^*}} b(z)$;

      Remove y from Σ_{ℓ^*} and add x to Σ_{ℓ^*};

      Mark y as unassigned and mark x as assigned;

      Set $d_{\ell^*} = \min(\min_{z \in \Sigma_{\ell^*}} b(z), 0)$;

    else

      Mark x as assigned and add x to Σ_{ℓ^*};

    end

  end

end

Set $\boldsymbol{p} = \max(\boldsymbol{d}, \boldsymbol{0})$, set $\boldsymbol{t} = \max(-\boldsymbol{d}, \boldsymbol{0})$;

return $(\boldsymbol{\Sigma}, \boldsymbol{p}, \boldsymbol{t})$

Proposition 4.1.1. *Given initial prices and incentives $\boldsymbol{p}^0, \boldsymbol{t}^0$, and an empty matching, Algorithm 4.3 produces an upper feasible ε-CS matching $\boldsymbol{\Sigma}$ with no over-incentivized phases with time complexity $O(NU(\log(U) + N)(C + G)/\varepsilon)$ where $U = \max_{1 \le \ell \le N} U_\ell$ and $G = \max_{1 \le \ell, j \le N}(p_j^0 - t_j^0) - (p_\ell^0 - t_\ell^0)$.*

Proof. Note that no phase can increase beyond U_ℓ members, and no phase can increase beyond B_ℓ members as long as $d_\ell < 0$. Therefore, the algorithm will not terminate until the matching is complete, upper feasible, and there are no over-incentivized phases. Throughout the auction the number of unmatched people is non-increasing and the variable $\boldsymbol{d}$ is entrywise non-decreasing. The monotonicity of these quantities allows us to use the same complexity argument as in Algorithm 4.1. The above bound will then immediately follow, where the factor G accounts for the prices and incentives not being initialized to zero.

It remains to show that the algorithm preserves ε-CS at every step. The only place where this algorithm differs from Algorithm 4.1 is when a person x wants to join a phase ℓ where $|\Sigma_\ell| = B_\ell$ and $d_\ell < 0$. Let $\boldsymbol{d}, \boldsymbol{d}'$ be the values before and after x is added. Since $d'_\ell \le \min_{y \in \Sigma_\ell} b(y)$ every person matched to ℓ must satisfy ε-CS. □

Algorithm 4.4: Lower Bound Auction

Input: $\varepsilon > 0$, bounds $\boldsymbol{B}, \boldsymbol{U}$, coefficients $\boldsymbol{a}$, initial prices $\boldsymbol{p}^0$, initial admissible incentives $\boldsymbol{t}^0$, complete (but possibly lower infeasible) ε-CS matching $\boldsymbol{\Sigma^0}$

Initialization: Set $\boldsymbol{d} = \boldsymbol{p}^0 - \boldsymbol{t}^0$, set $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^0$;

Result: complete and feasible ε-CS matching and admissible prices and admissible incentives $(\boldsymbol{\Sigma}, \boldsymbol{p}, \boldsymbol{t})$.

```
while there exists some ℓ with (|Σ_ℓ| < U_ℓ and d_ℓ > 0) or (|Σ_ℓ| < B_ℓ) do
    for each ℓ* with (|Σ_ℓ*| < U_ℓ* and d_ℓ* > 0) or (|Σ_ℓ*| < B_ℓ*) do
        for each x ∉ Σ_ℓ* do
            Let j be x's current phase;
            Calculate Δ(x) = (a_j(x) − d_j) − (a_ℓ*(x) − d_ℓ*);
        end
        while (|Σ_ℓ*| < U_ℓ* and d_ℓ* > 0) or (|Σ_ℓ*| < B_ℓ*) do
            Find x ∈ arg min_{y∉Σ_ℓ*} Δ(y);
            if |Σ_ℓ*| < B_ℓ* then
                Remove x from its current phase and add x to Σ_ℓ*;
                if |Σ_ℓ*| = B_ℓ* and Δ(x) ≥ 0 then
                    Subtract Δ(x) + ε from d_ℓ*;
                end
            else
                if Δ(x) + ε ≥ d_ℓ* then
                    Set d_ℓ* = 0;
                else
                    Remove x from its current phase and add x to Σ_ℓ*;
                    if |Σ_ℓ*| = U_ℓ* and Δ(x) ≥ 0 then
                        Subtract Δ(x) + ε from d_ℓ*;
                    end
                end
            end
        end
    end
end
Set p = max(d, 0), set t = max(−d, 0);
return (Σ, p, t)
```

Proposition 4.1.2. *Given the result of Algorithm 4.3, Algorithm 4.4 produces a complete*

and feasible ε-CS matching $\mathbf{\Sigma}$ with no overpriced or over-incentivized phases with time complexity $O(n^2N^2(C+G)/\varepsilon)$ where $G = \max_{\ell \neq j}(p_j^0 - t_j^0) - (p_\ell^0 - t_\ell^0)$.

Proof. It is clear that the algorithm will not terminate until the matching is complete and lower feasible, and there are no over-priced phases. The algorithm will never add people to an already full phase ℓ with $|\Sigma_\ell| = U_\ell$, thus the matching stays upper feasible. A phase only offers incentives if it has fewer than B_ℓ members, and any phase that has offered an incentive will never have more than B_ℓ members. Thus, no phase will become over-incentivized.

Next, we show that $\mathbf{\Sigma}$ is a complete ε-CS matching at every step of the algorithm. Consider what happens when a phase ℓ^* is modified. Let $(\mathbf{\Sigma}, \boldsymbol{d})$ the values before the modification and $(\mathbf{\Sigma}', \boldsymbol{d}')$ afterwards.

First, we consider the case where $|\Sigma_{\ell^*}| < B_{\ell^*}$. In this case Σ'_{ℓ^*} must now have B_ℓ points. Let x_f be the last point added to ℓ^*. If $d'_{\ell^*} = d_{\ell^*}$ then $\Delta(x_f) < 0$ and we can conclude that every person who had their membership switched to ℓ^* strictly preferred ℓ^* over their previous membership. Since no other entry of $\boldsymbol{d}$ changed, the new pair $(\mathbf{\Sigma}', \boldsymbol{d}')$ still satisfies ε-CS. Otherwise, $\Delta(x_f) \geq 0$ and $d'_\ell = d_\ell - \Delta(x_f) - \varepsilon$. Clearly everyone who was in Σ_ℓ is even happier to be in Σ'_{ℓ^*} as $d'_{\ell^*} < d_{\ell^*}$ and other entries of $\boldsymbol{d}$ didn't change. Next, we check the other people whose membership didn't change. Let y be some person $y \in \Sigma'_j$ for some $j \neq \ell^*$. We need to show that $\max_{1 \leq \ell \leq N} a_\ell(y) - d'_\ell - \varepsilon \leq a_j(y) - d'_j$. Only d'_{ℓ^*} is different so it is enough to show $a_{\ell^*}(y) - d'_{\ell^*} - \varepsilon \leq a_j(y) - d_j$. By our choice of x_f we have

$$a_{\ell^*}(y) - d'_{\ell^*} - \varepsilon = a_{\ell^*}(y) - d_{\ell^*} + \Delta(x_f) \leq a_{\ell^*}(y) - d_{\ell^*} + \Delta(y) = a_j(y) - d_j.$$

Finally we check the people who were switched to ℓ^*. Let z be one of those people and suppose that z was previously matched to phase r. Since $\Delta(x_f) \geq \Delta(z)$ we may conclude

$$\max_{\ell \neq \ell^*} a_\ell(z) - d'_\ell \leq a_r(z) - d_r + \varepsilon = a_{\ell^*}(z) - d_{\ell^*} + \varepsilon + \Delta(z) \leq a_{\ell^*}(z) - d'_{\ell^*}.$$

Next, we consider the case where $|\Sigma_{\ell^*}| < U_{\ell^*}$ and $d_{\ell^*} > 0$. This case is very similar, however there is one additional thing that can happen. Namely, it is possible that d'_{ℓ^*} can be set to zero before Σ'_{ℓ^*} reaches U_{ℓ^*} members. As before, let x_f be the last person added to ℓ^* in the modification, and let $y_c = \arg\min_{y \notin \Sigma'_\ell} \Delta(y)$. If x_f exists (possibly no one was added) then $\Delta(x_f) + \varepsilon < d_{\ell^*} \leq \Delta(y_c) + \varepsilon$. Similar arguments to the above now show that anyone in Σ'_{ℓ^*} satisfies ε-CS. To check that every other person satisfies ε-CS it is enough to

show that y_c satisfies ε-CS. Suppose that y_c is matched to a phase j. Then

$$a_{\ell^*}(y_c) - d'_{\ell^*} \leq a_{\ell^*}(y_c) + \Delta(y_c) + \varepsilon - d_{\ell^*} = a_j(y_c) - d_j + \varepsilon$$

which is enough to show ε-CS for y_c. Thus the algorithm preserves ε-CS.

Finally, we show that the algorithm terminates. Suppose for some ℓ the quantity d_ℓ decreases by more than $2(C+G)+\varepsilon$ from its starting value. Since $d^0_\ell - G \leq 0$ it must be the case that $|\Sigma_\ell| \leq B_\ell$. Immediately after d_ℓ is lowered to $d^0_\ell - C - 2G - \varepsilon$ phase ℓ must have exactly B_ℓ members. If the algorithm has not terminated then there must be some j with more than B_j members, and thus $d_j \geq d^0_j - G$. For any x we can then conclude that

$$a_\ell(x) - d_\ell - (a_j(x) - d_j) \geq a_\ell(x) - a_j(x) + d^0_j - d^0_\ell + 2C + G \geq 0.$$

It then follows that $|\Sigma_\ell| = B_\ell$ for the remainder of the auction, as it will always be easier for other phases to incentivize people to leave phase j rather than phase ℓ.

Notice that the same person cannot switch phases N times unless one of the entries of $\boldsymbol{d}$ has decreased. Thus, a phase ℓ can enter a bidding stage at most Nn times before d_ℓ must decrease by at least ε. This gives us an upper bound of $2N^2n\lceil(C+G)/\varepsilon\rceil$ bidding stages before the algorithm terminates. Quickselect (a variant of Hoare's quicksort algorithm) can be used to find the k smallest values of $\Delta(x)$ in time $O(n)$ regardless of k [45]. Thus the worst case complexity of the algorithm is $O(n^2N^2(C+G)/\varepsilon)$.

□

Both Algorithms 4.3 and 4.4 are compatible with ε scaling. The prices and incentives obtained from one iteration of Algorithms 4.3 and 4.4 together can be fed into the next iteration. For the instances of (4.18) that we encounter, the complexity of both algorithms using ε scaling appears to grow like $O(nN(\log(n)+N)\log(nC/\delta))$, where $\delta > 0$ is the maximum error of the final solution.

With the upper and lower bound auction algorithms in hand, we can now give the ver-

sion of auction dynamics with upper and lower volume bounds, Algorithm 4.5 below.

Algorithm 4.5: Auction Dynamics with Volume Bounds

Input: Domain D_n, initial configuration $\boldsymbol{\Sigma}$, surface tensions $\boldsymbol{\sigma}$, kernel K, volume bounds $\boldsymbol{B}, \boldsymbol{U}$, time step δt, number of steps m, auction error tolerance ε_{min}, epsilon scaling factor α, initial epsilon ε_0.

Result: Final configuration $\boldsymbol{\Sigma}^m$

Initialization: Set $\boldsymbol{\Sigma}^0 := \boldsymbol{\Sigma}$, set $\bar{\varepsilon} = \varepsilon_{min}/n$;

for k *from* 0 *to* $m-1$ **do**

  Calculate the convolutions: $\psi_\ell^{k+1}(x) = \sum_{j \neq \ell} \sigma_{\ell j}(K_{\delta t} * \Sigma_j^k)(x)$;

  Calculate the assignment problem coefficients: $\boldsymbol{a}^{k+1} = 1 - \boldsymbol{\psi}^{k+1}$;

  Initialize prices $\boldsymbol{p} = \mathbf{0}$, incentives $\boldsymbol{t} = \mathbf{0}$, and $\varepsilon = \varepsilon_0$;

  while $\varepsilon \geq \bar{\varepsilon}$ **do**

    Run Algorithm 4.3 (Upper Bound Auction):

    $(\boldsymbol{\Sigma}_{\text{out1}}, \boldsymbol{p}_{\text{out1}}, \boldsymbol{t}_{\text{out1}}) =$ Upper Bound Auction$(\varepsilon, \boldsymbol{B}, \boldsymbol{U}, \boldsymbol{a}^{k+1}, \boldsymbol{p}, \boldsymbol{t}, D_n)$;

    Run Algorithm 4.4 (Lower Bound Auction): $(\boldsymbol{\Sigma}_{\text{out2}}, \boldsymbol{p}_{\text{out2}}, \boldsymbol{t}_{\text{out2}}) =$

    Lower Bound Auction$(\varepsilon, \boldsymbol{B}, \boldsymbol{U}, \boldsymbol{a}^{k+1}, \boldsymbol{p}_{\text{out1}}, \boldsymbol{t}_{\text{out1}}, \boldsymbol{\Sigma}_{\text{out1}})$;

    Set $(\boldsymbol{p}, \boldsymbol{t}) = (\boldsymbol{p}_{\text{out2}}, \boldsymbol{t}_{\text{out2}})$;

    Divide ε by α;

    if $\varepsilon < \bar{\varepsilon}$ **then**

      Set $\boldsymbol{\Sigma}^{k+1} = \boldsymbol{\Sigma}_{\text{out2}}$;

    end

  end

end

return $\boldsymbol{\Sigma}^m$

4.1.4 Auction dynamics with temperature

Finally, we conclude this section with a variant of the auction dynamics algorithm that allows us to incorporate random fluctuations due to temperature. There are several reasons to introduce temperature effects into auction dynamics, two of these are: 1. In machine learning applications, temperature can help the algorithm escape from local minima and find better solutions. 2. Low temperature levels can be added to auction dynamics to help avoid degenerate auction coefficients (which slow down the algorithm) without significantly changing the result.

In the classic threshold dynamics algorithm, one may incorporate temperature in the style of rejection free Monte Carlo methods by randomizing the thresholding step. The

Monte-Carlo approach suggests randomly assigning each x to a phase i with probability:

$$\mathbb{P}(x \in \Sigma_i^{k+1}) = \frac{e^{-\beta \psi_i^{k+1}(x)}}{\sum_{j=1}^{N} e^{-\beta \psi_j^{k+1}(x)}}, \tag{4.22}$$

where $\beta = \frac{1}{T}$ is the inverse temperature. In the limit as $T \to 0$ one recovers the original MBO algorithm.

Unfortunately this approach is not compatible with auction dynamics. The volume constraints prevent us from assigning points independently. As a result, we cannot introduce the randomness in the assignment step. Instead, we introduce temperature before the assignment step by perturbing the coefficients $a_\ell(x) = (1 - \psi_\ell(x))$. Given some probability distribution $X = X(0, T)$ on the reals with mean zero and variance T, we perturb each coefficient $a_\ell(x)$ by an independent sample of X. This approach maintains the same basic properties as the randomness strategy (4.22). As $T \to 0$, we recover the original algorithm and as $T \to \infty$ the points are assigned to phases completely randomly. In our implementations of temperature effects, we choose the random variables to be normal random variables $N(0, T)$.

4.2 Curvature motion

We demonstrate the effectiveness of our auction dynamics code by computing several examples of volume preserving mean curvature motion in two and three dimensions. Since the focus of this work is to develop the necessary theory and algorithms for the volume constrained case, we work with essentially the most basic implementation of auction dynamics with the exception of the following well-known and simple trick to enhance the spatial resolution. The intermediate steps arising in each iteration of auction dynamics yields a smooth level set function (given by $\boldsymbol{\psi} - \boldsymbol{p}$) that can be used (via interpolation) to estimate the fraction of each grid cell occupied by a given phase. This allows for a sub-pixel accurate representation of the characteristic functions of the phases. For applications requiring greater efficiency or accuracy, one may turn to more sophisticated techniques developed for threshold dynamics, which in principle extend to auction dynamics as well.

We begin by considering two different equal volume tessellations of the torus. In Figure 4.1, the starting configuration is 64 randomly shifted squares of equal volume. After evolving under auction dynamics, the final configuration is a hexagonal lattice, which has optimal isoperimetric quotient among all equal volume tilings of the plane [41]. Thus, the algorithm finds the lowest energy state as one would hope. A more interesting example is

given in Figure 4.2. The starting configuration consists of 17 equal volume rectangles. In the case of 17 subunits, it is impossible to tile the torus with hexagons [59]. Indeed, the final configuration contains a heptagon and a pentagon. Nevertheless, most of the shapes are hexagons and visual inspection suggests that all of the triple junction angles are nearly 120 degrees. Therefore, the final configuration is a plausible minimizer of the interfacial perimeters.

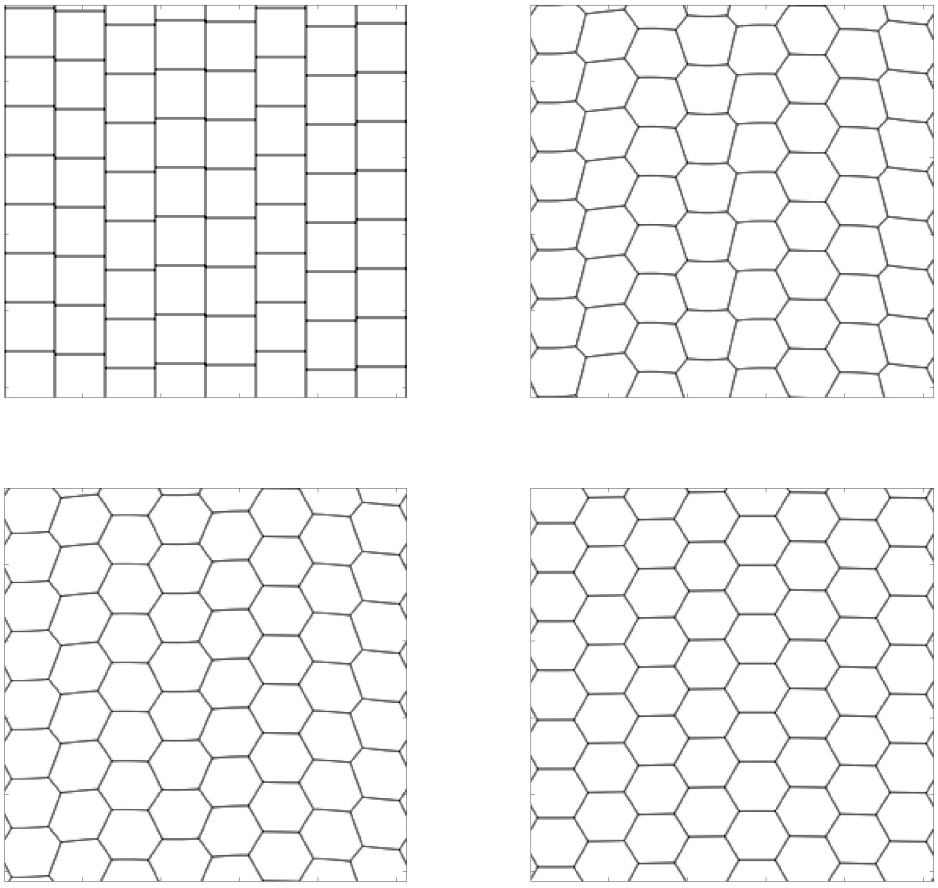

Figure 4.1: Initial condition: Randomly shifted 8 columns of 8 squares that have identical areas. Periodic boundary conditions.

Next, we consider random Voronoi diagrams in 2 and 3 dimensions. Figure 4.3 depicts the evolution of a random Voronoi diagram in the plane. The network immediately undergoes topological changes – all of quadruple junctions in the initial configurations split and form triple junctions. Figure 4.4 shows the evolution of a single "grain" in a random Voronoi diagram in 3 dimensions, Figure 4.5 shows the same grain and several of its neighbors at the final time. One can clearly see many topological changes in the faces of

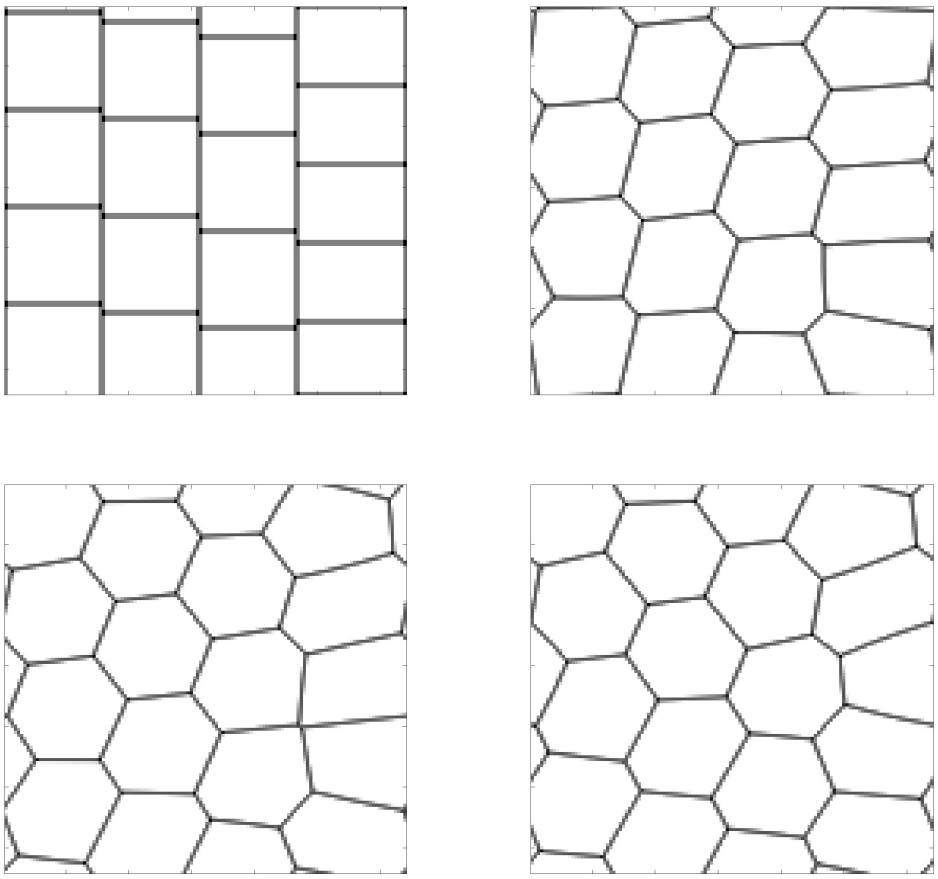

Figure 4.2: Initial condition: Randomly shifted 17 rectangles that have identical areas. Periodic boundary conditions. After a long time, there is still one phase with five and another with seven neighbors.

the grain. Quadruple junctions split and collide throughout the evolution. Both examples clearly show that one must anticipate topological changes in the course of the flow.

Finally, we consider equal volume tilings in 3 dimensions. Our starting configuration is a randomly shifted cubic lattice with 8 phases. Unlike the two dimensional case above, where the flow easily found the optimal solution, the 3 dimensional energy landscape appears to be littered with local minima. Regardless of how the cubes are shifted, the configuration evolves to a final state where each grain assumes the shape shown in Figure 4.6 – a 12 sided polytope built from 4 trapezoids, 4 rhombi, and 4 hexagons. A simple calculation shows that the isoperimetric quotient of this structure is considerably worse than several well-known tilings of 3-space. On the other hand, if we run the flow in the presence of temperature, the random fluctuations allow us to escape the local minima. Figure 4.7 shows an experiment with temperature where the final configuration assumes the structure

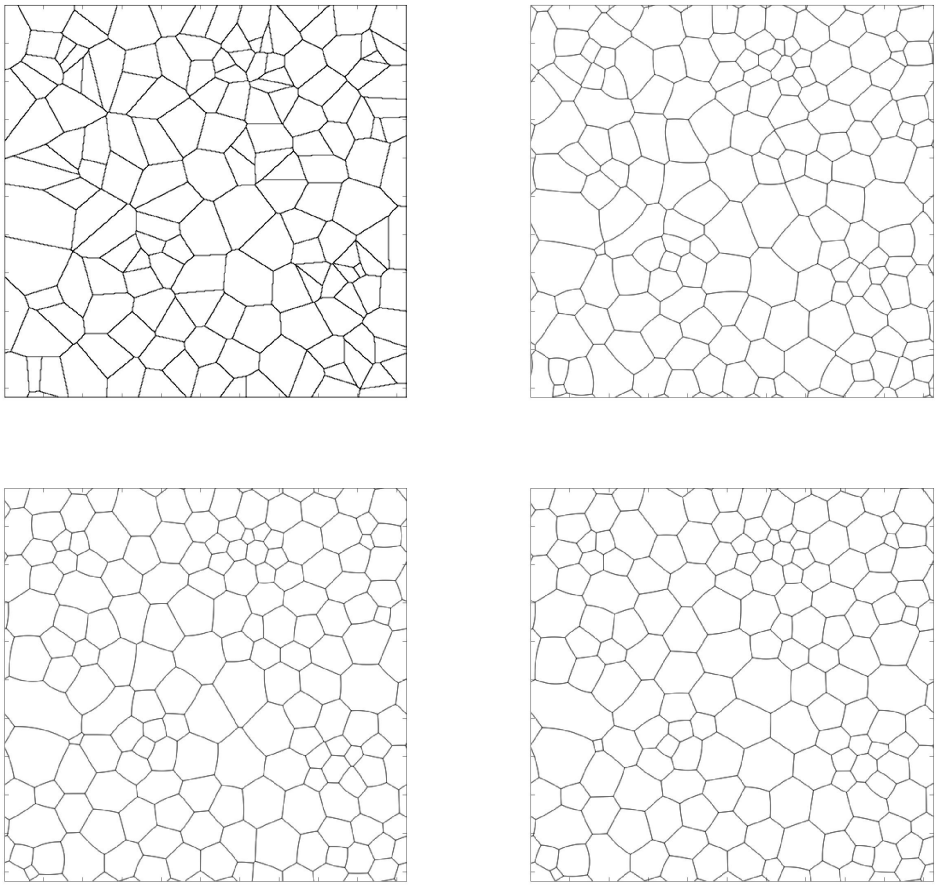

Figure 4.3: Initial condition: Voronoi diagram of 160 points taken uniformly at random on $[0,1]^2$. Periodic boundary conditions. Each phase preserves its initial area.

of what is thought to be the most efficient partition of 3-space, the Weaire-Phelan structure [77]. This experiment suggests that auction dynamics with temperature may be a very useful tool for exploring minimal tilings in 3 dimensions.

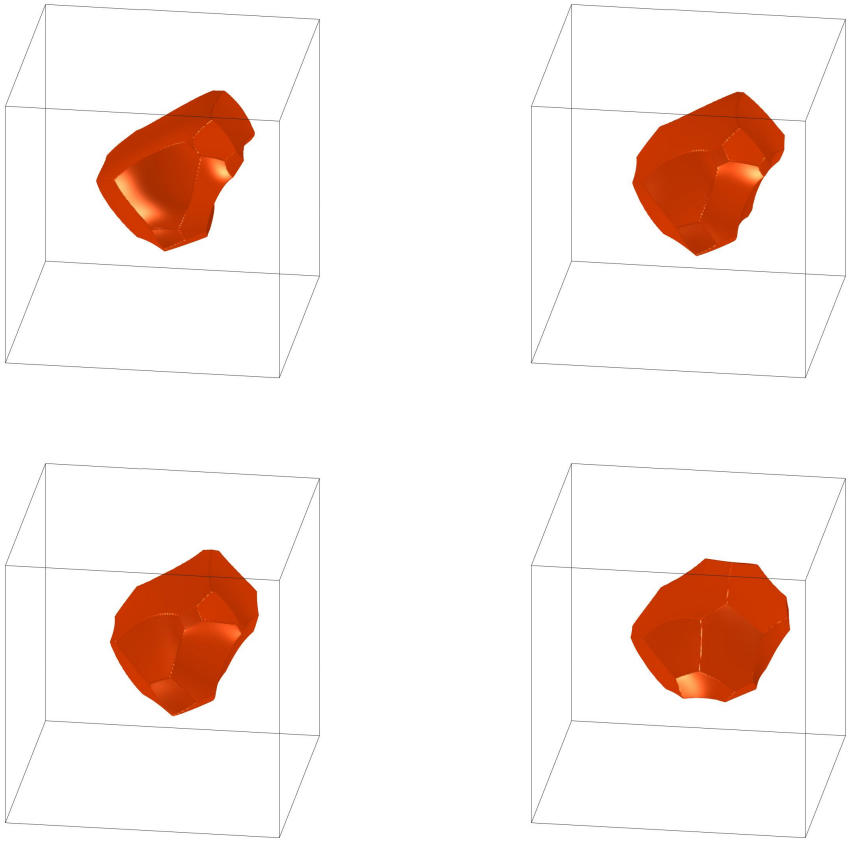

Figure 4.4: One "grain" from a total of 32. Initial condition: Voronoi diagram of 32 points taken uniformly at random on $[0,1]^3$. Periodic boundary conditions. Each phase preserves its initial volume.

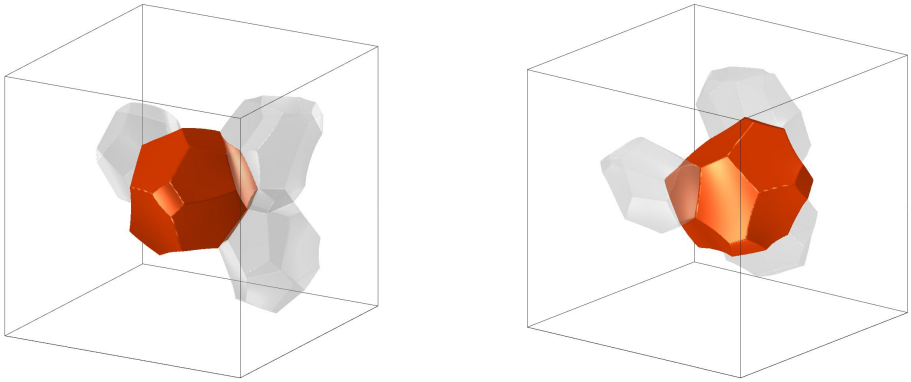

Figure 4.5: At final time, from a couple of other angles, with a few of its neighbors showing.

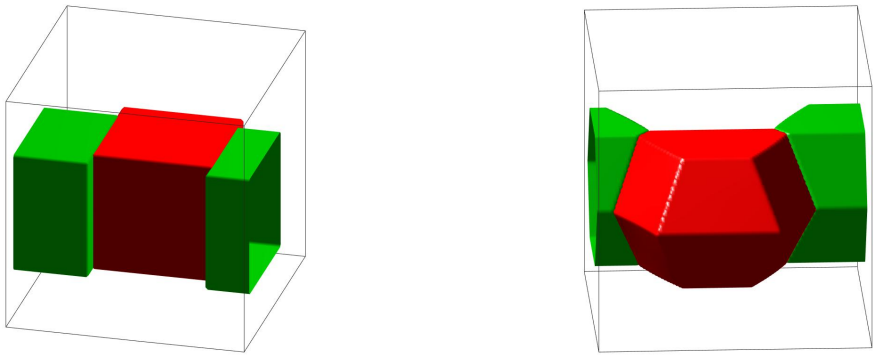

Figure 4.6: The initial and final configurations of the volume preserving flow on a randomly shifted cubic lattice. Each image shows two of the grains. The final configuration is fixed under the flow, but is not the global minimizer of the surface energy.

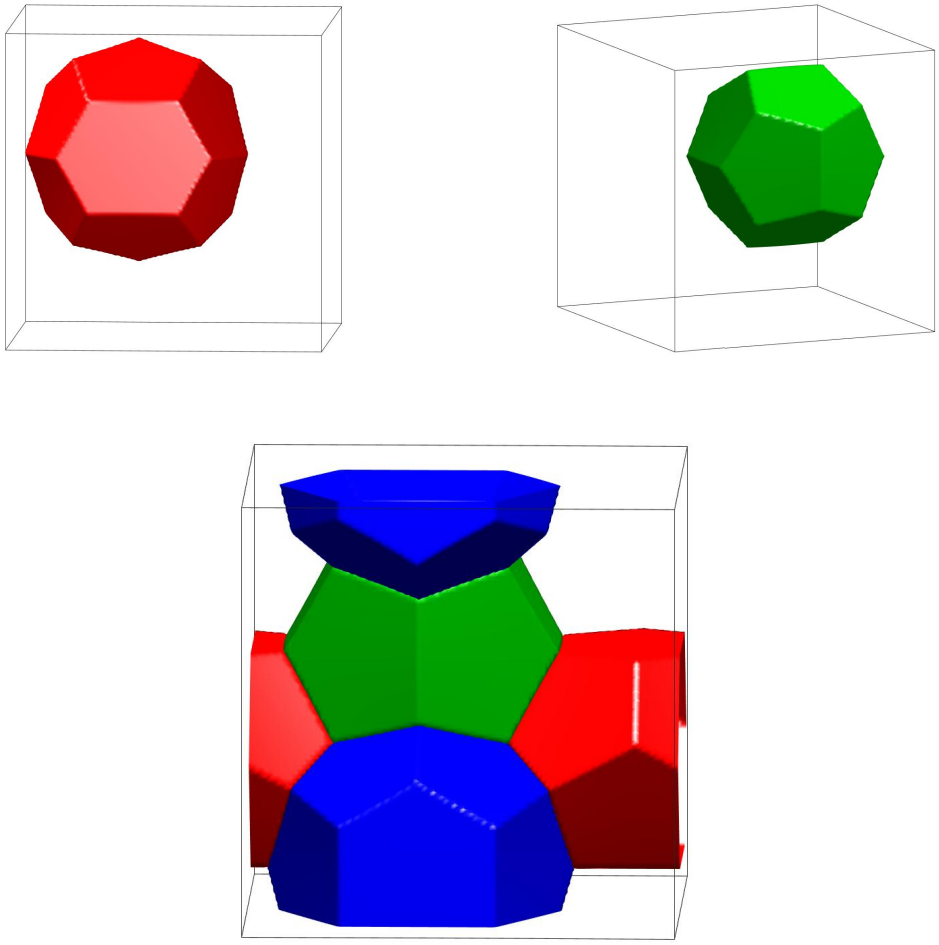

Figure 4.7: Running the flow on the 8 subunit cubic lattice with temperature fluctuations leads to the Weaire-Phelan structure. The Weaire-Phelan structure contains two distinct subunits shown in the first two images, the truncated hexagonal trapezohedron on the left and the pyritohedron on the right. The bottom image shows how 3 of the subunits fit together.

CHAPTER 5

Semi-Supervised Learning [4]

5.1 Introduction

Classifying high dimensional data is one of the central problems in machine learning and computer vision. The graphical approach to these problems builds a weighted graph from the data set and searches for an optimal partitioning of the vertices into distinct classes. The search is driven by the goal of minimizing the total weight of cut edges between adjacent vertices in different classes. To avoid trivial solutions, it is necessary to impose certain constraints or penalties on the segmentations. For example, one may penalize solutions that do not give a reasonably uniform distribution of vertices among the different classes. In general, solving graph partitioning problems with combinatorial penalties, such as the normalized cut [57] or Cheeger cut [19], is known to be NP-hard. The essential difficulty stems from the fact that one is attempting to minimize a non-convex objective function. Nonetheless, approximate solutions have been calculated using spectral clustering (for example [57], [80]), and more recently fast implementations of gradient descent [42], [16].

In this chapter we consider the SSL data classification problem. In the SSL setting, the number of classes is known and a certain training subset of the data is provided with the ground truth classification given. The objective is to then classify the remaining points using the training data. The SSL problem is highly amenable to variational methods. The training data can be incorporated into norm or linear type penalty terms that are much easier to solve than the combinatorial penalties of the unsupervised methods mentioned above. Recent results in SSL have shown that variational methods are competitive with artificial neural networks, while requiring far less training data and computation time to obtain high quality solutions [33].

The natural connection between threshold dynamics and minimal partition problems

[4] This Chapter combines [49] with some machine learning results from [50] and [26]. [49] has been accepted to the Sixth International Conference on Scale Space and Variational Methods in Computer Vision.

has led to several MBO type schemes for solving segmentation problems in machine learning and computer vision. The authors of [28] derived an MBO scheme from the Ginzburg-Landau energy to solve the piecewise constant Mumford-Shah functional. Building on the approach of [28], in a series of papers [8, 61, 33], Bertozzi et al. introduced a binary and then multiclass version of the Ginzburg-Landau energy on graphs, and derived MBO schemes for the SSL problem.

In this chapter, we will adapt the MBO schemes and variants from the previous chapters to the graphical setting, and use them to solve the SSL problem. The key to our approach will be the graph heat content (GHC), a graph variant of the heat content energy. The natural extension of the heat content to graphs allows us to easily translate our previous algorithms to graphs. Furthermore, the resulting algorithms will immediately inherit many of the beneficial properties from our schemes above. Most importantly, we will be able to guarantee energy dissipation and unconditional stability of the algorithm for an extremely wide class of graph diffusion processes. This represents a significant theoretical advantage over other recent graph MBO schemes [33, 60] for the SSL problem. Our most straightforward SSL algorithms, based on the MBO schemes of Chapter 2, produce comparable or superior results to other state-of-the-art variational methods, while running several orders of magnitude faster. Our more sophisticated algorithms, based on the volume constraint and temperature modifications of Chapter 4, are effective for unprecedentedly low levels of training data and still run faster than other state-of-the-art methods.

5.2 Background and Notation

5.2.1 The graphical model

We consider the SSL data classification problem over the structure of an undirected weighted graph $G = (\mathcal{V}, W)$. $\mathcal{V}$ is the set of data points, and the weight matrix $W : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is a symmetric matrix that describes the connection strength between any two points.

The datasets we consider in this work are collections of real vectors embedded in a high dimensional Euclidean space. A key assumption of machine learning is that the data is concentrated near a low dimensional manifold. Our goal is to reflect this manifold structure in our choice of weight matrix. Ideally, we would like to weight points based on the geodesic distances between them, however this information is not readily available to us and would lead to a very dense weight matrix. Instead, we assume that the manifold is locally Euclidean, and only compute the k nearest neighbors of each point in the Euclidean metric. Computing just a small fraction of the distances ensures that W will be a sparse

matrix, which will be essential for the fast performance of our algorithms.

Under these assumptions a popular choice for the weights are the Zelnick-Manor and Perona (ZMP) weight functions [81]:

$$W(x,y) = \exp\big(\frac{-d_E(x,y)^2}{\sigma(x)\sigma(y)}\big) \tag{5.1}$$

where d_E is the Euclidean distance and $\sigma(x), \sigma(y)$ are local scaling parameters for x, y respectively. We will construct our weight matrices using the ZMP weights, where we set $\sigma(x) = d_E(x, x_r)$ where x_r is the r^{th} nearest neighbor of x. To recover a symmetric matrix we simply set $W(x,y) \leftarrow \max(W(x,y), W(y,x))$.

It will be useful for us to have a notion of an approximate geodesic distance between points in the graph that are not nearest neighbors. With the structure of the weight matrix, we may compute approximations to the geodesic distance by traversing through paths in the graph. Let a path p in the graph be a sequence of vertices $\{x_1, \dots, x_s\}$ such that $W(x_i, x_{i+1}) \neq 0$ for every $1 \leq i \leq s-1$. Let the length $\ell_q(p)$ of a path be

$$\ell_q(p) = \Big(\sum_{1\leq i\leq s-1} (-\log(W(x_i,x_{i+1})))^{q/2}\Big)^{1/q} = \Big(\sum_{1\leq i\leq s-1} \big(\frac{d_E(x_i,x_{i+1})}{\sqrt{\sigma(x)\sigma(y)}}\big)^q\Big)^{1/q} \tag{5.2}$$

Let $\pi(x,y)$ be the set of all paths from x to y. Then the q-geodesic distance between x and y, denoted $d_{G,q}(x,y)$, may be defined as

$$d_{G,q}(x,y) = \min_{p\in\pi(x,y)} \ell_q(p) \tag{5.3}$$

Given any subset $S \subset \mathcal{V}$ the distances $d_{G,q}(x,S) = \min_{z\in S} d_{G,q}(x,z)$ may be efficiently computed using Dijkstra's algorithm [24].

5.2.2 Semi-supervised data classification

Given a set of data points $\mathcal{V}$, a fixed collection of labels $\{1, \dots, N\}$, and a fidelity subset $F \subset \mathcal{V}$ of points whose labels are known, the semi-supervised data classification problem asks to correctly label the remaining points in $\mathcal{V} \setminus F$. Any solution of the problem is a partition $\mathbf{\Sigma} = (\Sigma_1, \dots, \Sigma_N)$ of $\mathcal{V}$ where Σ_i is the set of points that are assigned label i. An N-phase partition of $\mathcal{V}$ may be represented as a function $u : \mathcal{V} \to \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ where $\mathbf{e}_i \in \mathbb{R}^N$ is the i^{th} standard basis vector. The convex relaxation of this space is the set of

functions $u : \mathcal{V} \to \mathcal{S}_N$, where $\mathcal{S}_N$ is the simplex

$$\mathcal{S}_N = \{\mathbf{p} \in [0,1]^N : \sum_{i=1}^{N} p_i = 1\} \tag{5.4}$$

A point $\mathbf{p} \in \mathcal{S}_N$ can be interpreted as a vector of probabilities, where p_i gives the confidence that a point should be assigned label i. We will denote the ground truth segmentation of the points as the function $\mathbf{u}^*$.

Variational approaches solve the problem by finding minimizers of energies of the form

$$E(\mathbf{u}) = R(\mathbf{u}) + \text{Fid}(\mathbf{u}). \tag{5.5}$$

Here R is a regularizing term that is typically some relaxation of the weighted graph cut (5.6), and Fid is a term that incorporates the fidelity data F.

$$\text{Cut}(\mathbf{\Sigma}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{x\in\Sigma_i}\sum_{y\notin\Sigma_i} W(x,y). \tag{5.6}$$

Given some constants $f_i(x)$, we will assume throughout that $\text{Fid}(\mathbf{u})$ has the linear structure (5.7). We will address the connection between the coefficients $f_i(x)$ and the fidelity set F in Section 5.3.2.

$$\text{Fid}(\mathbf{u}) = \sum_{i=1}^{N}\sum_{x\in\mathcal{V}} f_i(x)u_i(x) \tag{5.7}$$

5.3 Graph MBO Schemes

There are many possible relaxations of the weighted graph cut (5.6). Our approach is to model the graph cut with the graph heat content energy (GHC). GHC is a family of energies indexed by diffusion matrices, symmetric matrices $A : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, which play the role of the convolution kernel K in the graph setting. Given a diffusion matrix A, the graph heat content of a function $\mathbf{u} : \mathcal{V} \to \mathcal{S}_N$ is

$$\text{GHC}(\mathbf{u}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{x,y\in\mathcal{V}} A(x,y)u_i(x)(1-u_i(y)). \tag{5.8}$$

If the affinity matrix A is the weight matrix W, then GHC is a relaxation of the graph cut. One can also define more complex versions of GHC in the vein of energy (2.25), where each interface $\partial\Sigma_i \cap \partial\Sigma_j$ has a different associated diffusion matrix A_{ij}. However, in the

context of graph segmentation problems it is not clear how to choose or learn an appropriate network of diffusion matrices $\boldsymbol{A}$. Thus, we will leave the investigation of the more general graph energy for a future work, and focus on energy (5.8).

5.3.1 MBO via linearizations of GHC

We now derive an MBO scheme for minimizing SSL energies of the form:

$$E(\mathbf{u}) = \text{GHC}(\mathbf{u}) + \text{Fid}(\mathbf{u}). \tag{5.9}$$

Recall that the authors of [27] recovered the MBO scheme by minimizing linearizations of the heat content. We apply this same approach to GHC to obtain the graph analogue of the MBO scheme. To that end, consider the variation of GHC at a configuration $\mathbf{u}$ in the direction of $\boldsymbol{\varphi}$. Using the quadratic structure of GHC we get:

$$\text{GHC}(\mathbf{u}+\boldsymbol{\varphi}) = \text{GHC}(\mathbf{u}) + \frac{1}{2}\sum_{i=1}^{N}\sum_{x\in\mathcal{V}}\varphi_i(x)\sum_{y\in\mathcal{V}}A(x,y)\Big(\big(1-2u_i(y)\big)-\varphi_i(y)\Big). \tag{5.10}$$

When A is positive semi-definite (PSD), the quadratic form $-\varphi_i^T A\varphi_i$ is negative for all $\varphi_i : \mathcal{V} \to \mathbb{R}$, thus we may conclude:

$$\text{GHC}(\mathbf{u}+\boldsymbol{\varphi}) - \text{GHC}(\mathbf{u}) \leq \frac{1}{2}\sum_{i=1}^{N}\sum_{x\in\mathcal{V}}\varphi_i(x)\sum_{y\in\mathcal{V}}A(x,y)(1-2u_i(y)). \tag{5.11}$$

Combining (5.11) with energy (5.9) and taking advantage of the linear structure of Fid, we obtain:

$$E(\mathbf{u}+\boldsymbol{\varphi}) - E(\mathbf{u}) \leq \text{Fid}(\boldsymbol{\varphi}) + \frac{1}{2}\sum_{i=1}^{N}\sum_{x\in\mathcal{V}}\varphi_i(x)\sum_{y\in\mathcal{V}}A(x,y)(1-2u_i(y)). \tag{5.12}$$

The right hand side of equation (5.12) is the linearization of (5.9) at the function $\mathbf{u}$. The inequality implies that we may obtain a configuration of lower energy, $\mathbf{u}+\boldsymbol{\varphi}$, by minimizing the linearization over valid directions $\boldsymbol{\varphi}$. The only constraint on $\boldsymbol{\varphi}$ is that $\mathbf{u}+\boldsymbol{\varphi}$ must be an element of the domain of E, i.e. $\mathbf{u}(x)+\boldsymbol{\varphi}(x) \in \mathcal{S}_N$ for all x. This allows us to easily solve the right hand side of (5.11), and as expected, we see that the minimizer $\boldsymbol{u}+\boldsymbol{\varphi}$ is a partition where each phase is given by the thresholding operation:

$$\Sigma_i = \{x \in \mathcal{V} : i = \underset{1\leq j\leq N}{\arg\min}\, f_j(x) - \psi_j(x)\} \tag{5.13}$$

where $\psi_j(x) = \sum_{y \in \mathcal{V}} A(x,y) u_j(y)$ is the diffusion value of u_j at x. Iterating (5.12) and (5.13), we obtain Algorithm 5.1, GHCMBO, our first MBO algorithm for solving the SSL problem. Each iteration of GHCMBO dissipates (5.9) and the configuration space is compact, thus the algorithm must converge to a fixed point. In fact, fixed points of (5.11) are necessarily local minima of the energy (where we give the space of functions $\boldsymbol{u} : \mathcal{V} \to \mathcal{S}_N$ the topology of $\mathcal{S}_N^{|\mathcal{V}|}$). The guarantee of energy dissipation and convergence represents a significant theoretical advancement over previous graph MBO schemes for the SSL problem [33], [60]. The properties of GHCMBO that we have discovered above are summarized in Proposition 5.3.1.

Proposition 5.3.1. *If the given diffusion matrix A is PSD then Algorithm 5.1, GHCMBO, dissipates energy (5.9) at every step and converges to a fixed point in a finite number of steps.*

Algorithm 5.1: GHCMBO

The $(n+1)^{\text{th}}$ partition $\boldsymbol{\Sigma}^{n+1}$ is obtained from the n^{th} partition $\boldsymbol{\Sigma}^n$ as follows:

1. Diffusion by A:

$$\psi_i^{n+1}(x) = \sum_{y \in \Sigma_i^n} A(x,y) \quad \text{for} \quad 1 \le i \le N \tag{5.14}$$

2. Thresholding:

$$\Sigma_i^{n+1} = \{x \in \mathcal{V} : i = \underset{1 \le j \le N}{\arg\min}\, f_j(x) - \psi_j^{n+1}(x)\} \quad \text{for} \quad 1 \le i \le N \tag{5.15}$$

In addition to the favorable theoretical properties, GHCMBO is extremely fast. At every step the configuration is a partition, thus computing the vector $\boldsymbol{\psi}^{n+1}(x) = \left(\psi_1^{n+1}(x), \ldots, \psi_N^{n+1}(x)\right)$ requires just $\deg_0(x)$ additions, where $\deg_0(x)$ counts the number of nonzero entries of A in row x. As a result, when A is sparse, each iteration has low computational complexity. Furthermore, the step sizes of the scheme are very large, allowing for rapid convergence. The combination of simple computations and large step sizes makes GHCMBO significantly faster than other state-of-the-art methods (cf. timings in Tables 5.2-5.6).

To adapt GHCMBO to the problem at hand, we need to construct a PSD affinity matrix A that is related to the weighted graph structure $G = (\mathcal{V}, W)$. The simplest choice is to take $A = W^2$. Another possible choice is the graph heat kernel $H_t = e^{-tL}$, where L is the symmetric normalized graph Laplacian and $t > 0$. However, this adds a parameter t, and the heat kernel is typically not sparse. Previous graph MBO schemes [33], [60] have been

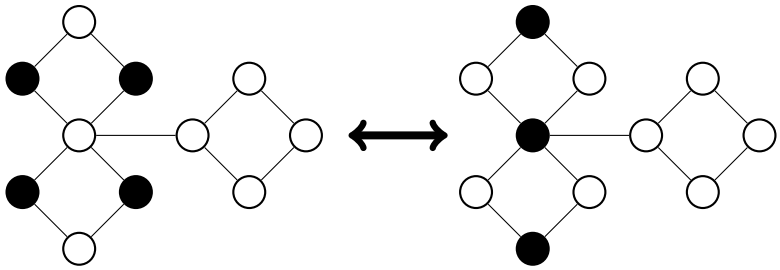

Figure 5.1: Behavior of Algorithm 5.1 under the action of the weights A_1 on a two-phase configuration with no fidelity term. The algorithm gets trapped in a periodic loop between the two configurations above. The configuration on the right has a higher energy.

restricted to diffusion by the heat equation and associated kernels. In addition to energy dissipation, one of the chief advantages of our approach is the ability to more freely choose a diffusion generated by a sparse matrix while avoiding extra parameters.

A natural question to ask is when can W itself be chosen for A. W is a desirable choice, as W is the sparsest matrix that still retains the full structure of the graph. Furthermore, when $A = W$ the graph heat content is a relaxation of the weighted graph cut. In general, one cannot expect that W as constructed in (5.1) will be positive semi-definite. Unfortunately, if the diffusion A has negative eigenvalues, GHCMBO may increase the energy and get stuck in an infinite loop as the following example shows.

Example 3:

Let A' be the adjacency matrix of the graph in Figure 5.1, I the identity matrix and take $A_1 = A' + \lambda I$ for any $\lambda < 2$. Applying Algorithm 5.1 to the graph (without a fidelity term) causes it to get trapped in a periodic loop between the two configurations. The left hand side has 8 edges between vertices in different classes while the right hand side has 9 edges between vertices in different classes. Thus, the algorithm increases the energy every time it moves from left to right.

While we cannot expect GHCMBO to be unconditionally stable in the case $A = W$, Chapter 2 suggests that we can work with a single growth variant instead. Indeed in the case that $A = W$, the entries of A will be non-negative, which is a sufficient condition for the convolution kernel in Propositions 2.2.5 and 2.2.6. Each interface has the same diffusion matrix A, thus, we will construct a graph analogue of the simpler single growth algorithm, Algorithm 2.4. We twist the construction slightly, and instead consider a single shrink scheme, which has a more efficient implementation. The single shrink graph scheme, GHCMBOS, is presented in Algorithm 5.2 below.

To prove that GHCMBOS dissipates energy (5.9) we only need to show that the quantity $-\sum_{i=1}^{N} \varphi_i^T A \varphi_i$ is negative for perturbation directions $\boldsymbol{\varphi}$ that arise in the single shrink

Algorithm 5.2: GHCMBOS

The $(n+1)^{\text{th}}$ partition $\mathbf{\Sigma}^{n+1}$ is obtained from the n^{th} partition $\mathbf{\Sigma}^n$ by a sequence of substeps $\mathbf{\Sigma}^{n,\ell}$ indexed by $\ell \in \{1, \dots, N\}$. Define $\mathbf{\Sigma}^{n,0} := \mathbf{\Sigma}^n$ and $\mathbf{\Sigma}^{n+1} := \mathbf{\Sigma}^{n,N}$, then $\mathbf{\Sigma}^{n,\ell}$ is obtained from $\mathbf{\Sigma}^{n,\ell-1}$ as follows:

1. Diffusion by A:

$$\psi_i^{n,\ell}(x) = \sum_{y \in \Sigma_i^{n,\ell-1}} A(x,y) \quad \text{for} \quad 1 \le i \le N \tag{5.16}$$

2. Restricted Thresholding:

$$\Sigma_\ell^{n,\ell} = \{x \in \Sigma_\ell^{n,\ell-1} : \ell = \operatorname*{arg\,min}_{1 \le j \le N} f_j(x) - \psi_j^{n,\ell}\} \tag{5.17}$$

$$\Sigma_i^{n,\ell} = \Sigma_i^{n,\ell-1} \cup \{x \in \Sigma_\ell^{n,\ell-1} : i = \operatorname*{arg\,min}_{1 \le j \le N} f_j(x) - \psi_j^{n,\ell}\} \quad \text{for} \quad i \ne \ell \tag{5.18}$$

scheme. In the scheme, each phase is either growing or shrinking but never both. Therefore the perturbations φ_i are all entrywise nonnegative or nonpositive. Thus, it suffices for A to have the form $A = A^1 + A^2$ where A^1 is PSD and A^2 is entrywise nonnegative. A further adaptation of the arguments of Propositions 2.2.5 and 2.2.6 immediately give Proposition 5.3.2.

Proposition 5.3.2. *If the given diffusion matrix A may be written as a sum $A = A^1 + A^2$ where A^1 is PSD and A^2 is entrywise nonnegative then Algorithm 5.2, GHCMBOS, dissipates energy (5.9) at every step and converges to a fixed point in a finite number of steps. Furthermore, if the diagonal entries $A(x,x)$ are strictly positive then any fixed point of GHCMBOS is also a fixed point of GHCMBO.*

Although GHCMBOS appears to require more computation than GHCMBO, the increase in complexity is modest. At the ℓ^{th} substep, calculations (5.16-5.18) are only necessary for $x \in \Sigma_\ell^{n,\ell-1}$. Thus, the complexity of a full step of GHCMBOS is comparable to the complexity of a step of GHCMBO. In our experiments GHCMBOS runs faster than GHCMBO (cf. Tables 5.2-5.6). The sparsity of W as compared to W^2 offsets any potential increase in complexity.

5.3.2 A fidelity term based on graph geodesics

Thus far, we have not described how to construct $\text{Fid}(\mathbf{u}) = \sum_{i=1}^N \sum_{x \in \mathcal{V}} f_i(x) u_i(x)$ from the fidelity data F. The simplest way is to impose a penalty on points whose labeling differs

from the ground truth labeling, $\mathbf{u}^*$, on F. Thus, we may take $f_i(x) = \lambda(1 - u_i^*(x))$ for $x \in F$ and zero for all other x. When λ is taken to infinity, the fidelity term becomes a hard constraint. We can easily incorporate the hard constraint into the minimization algorithms GHCMBO and GHCMBOS by simply not updating the points in the fidelity set.

If $\text{Fid}(\mathbf{u})$ is only active on fidelity nodes, the ground truth labeling $\mathbf{u}^*$ may be difficult to find in the energy landscape, especially when the size of F is very small compared to $\mathcal{V}$. For example, if F is small, then the global minimum of the energy will be near a partition that assigns all points outside of the fidelity set to the same label. For this reason, we introduce a fidelity term that is active on all of the nodes. Our approach is inspired by the region force in [79]. There, the authors introduce a linear penalty term where $f_i(x)$ is based on the diffusion distance [21] between x and elements of the fidelity set with labeling i.

Our fidelity term instead uses the graph geodesic distance defined in equation (5.3). For nodes in the fidelity set we use the hard constraint described above. For $x \notin F$, and some positive constant τ we take

$$f_i(x) = -\tau \exp\big(-d_{G,2}(x, F_i)^2\big). \tag{5.19}$$

where F_i is the set of fidelity points labeled i. We find that our fidelity term outperforms the diffusion distance fidelity term of [79]. On the MNIST data set, the initialization produced by labeling $x \in \mathcal{V} \setminus F$ according to $i(x) = \arg\min_j f_j(x)$ is much closer to the ground truth labeling, when using (5.19) instead of the term in [79] (see Table 5.4).

5.3.3 Schemes with volume constraints and temperature

In addition to a fidelity term, one may improve the accuracy of the variational model (5.9) by imposing volume constraints on each of the phases. Volume constraints are useful as they help the algorithm avoid trivial solutions where nearly all of non-fidelity vertices are assigned the same label. While it may be difficult to obtain the exact sizes of each class, it is generally possible to estimate a reasonable range using the fidelity data or low-level prior knowledge about the data set. Thus, our goal is to minimize energy (5.9) with the additional constraints $B_i \leq |\Sigma_i| \leq U_i$ for some set of upper and lower bounds $\boldsymbol{U}, \boldsymbol{B}$.

Following the (by now familiar) approach of minimizing linearizations of the proposed energy, we are led to solve

$$\arg\min_{\boldsymbol{u}:\mathcal{V}\to\mathcal{S}_N} \sum_{i=1}^{N} \sum_{x\in\mathcal{V}} u_i(x)(f_i(x) - \psi_i(x)) \quad \text{s.t.} \quad B_i \leq \sum_{x\in\mathcal{V}} u_i(x) \leq U_i \tag{5.20}$$

where $\psi_i(x) = \sum_{y \in \mathcal{V}} A(x,y) u_i(y)$. As we saw in Chapter 4, problem (5.20) may be solved using auction algorithms. The only difference is in the computation of the assignment coefficients. Therefore, we may use the auction algorithms of Chapter 4 essentially unchanged. This immediately gives us an MBO algorithm for solving graph partitioning problems with volume constraints, GHCMBOV, given in Algorithm 5.3 below.

Algorithm 5.3: GHCMBOV

The $(n+1)^{\text{th}}$ partition $\mathbf{\Sigma}^{n+1}$ is obtained from the n^{th} partition $\mathbf{\Sigma}^n$ as follows:

1. Diffusion by A:

$$\psi_i^{n+1}(x) = \sum_{y \in \Sigma_i^n} A(x,y) \quad \text{for} \quad 1 \leq i \leq N \tag{5.21}$$

2. Calculate assignment coefficients:

$$a_i^{n+1}(x) = \psi_i^{n+1}(x) - f_i(x) \quad \text{for} \quad 1 \leq i \leq N \tag{5.22}$$

3. Solve the assignment problem (see Algorithms 4.2 or 4.5):

$$\mathbf{\Sigma}^{n+1} = \underset{\boldsymbol{u}:\mathcal{V} \to \mathcal{S}_N}{\arg\max} \sum_{i=1}^{N} \sum_{x \in \mathcal{V}} a_i^{n+1}(x) u_i(x) \quad \text{s.t.} \quad B_i \leq \sum_{x \in \mathcal{V}} u_i(x) \leq U_i \tag{5.23}$$

It follows from our previous arguments that GHCMBOV dissipates energy (5.9) at every step as long as A is PSD. Of course, we would like to relax the conditions on A needed for energy dissipation. Naturally, this would entail extending the single growth schemes to the current situation. Unfortunately, single growth schemes are fundamentally incompatible with volume constrained problems. For most configurations, the solution to the assignment problem involves membership swaps that cannot be attained if certain phases must strictly grow or shrink. Thus, we must confine ourselves to working with PSD diffusion matrices.

While we cannot accommodate single growth schemes, there is no difficulty adding temperature to GHCMBOV as described in Chapter 4. Temperature is especially useful as it helps our (non-convex) algorithm escape from local minima. We find that adding temperature significantly improves the accuracy of GHCMBOV, especially at low fidelity percentages. As a result, our algorithm remains effective at fidelity percentages that are too low for other SSL methods. Temperature may also be added to algorithms GHCMBO and GHCMBOS, however it must be done with great care – the temperature cannot be set too high. In the absence of constraints, temperature may drive the algorithms to trivial

Table 5.1: Benchmark datasets

Dataset	Dimension	Points	Classes	W construction timing (s)
Three Moons	100	1,500	3	0.025
MNIST	784	70,000	10	149.04
Opt-Digits	64	5,620	10	2.03
COIL	241	1,500	6	0.33

solutions where nearly all vertices have the same label.

5.4 Experimental Results

We test our algorithms on several benchmark machine learning datasets: Three Moons, MNIST, Opt-Digits, and COIL. All experiments were run using C code on a single core of an Intel i5-4250U processor at 1.30 GHz with 4GB RAM. k-nearest neighbors were calculated using the kd-tree code in the VLFeat library. Table 5.1 shows the timing information for VLFeat. All of our subsequent timing results for GHCMBO, GHCMBOS and GHCMBOV include the time it takes to calculate the fidelity coefficients $f_i(x)$ and run the iterations (5.14-5.15), (5.16-5.18), or (5.21-5.23). On all datasets, we build the weight matrix using the ZMP weights (5.1), and choose the nearest neighbor and weight matrix scaling parameters k and r experimentally. To the best of our knowledge, there is no simple and principled way of choosing these values beyond experimentation. Choosing suboptimal k and r has a modest impact – about a 0.2-1.5% drop in accuracy.

5.4.1 Benchmark datasets

Here, we give a brief introduction to the various benchmark machine learning datasets that we tested against. We use the same weight matrix construction (detailed for each dataset below) in all experiments.

5.4.1.1 Three Moons

The Three Moons synthetic data set consists of three half circles embedded into $\mathbb{R}^{100}$ with Gaussian noise. The standard construction is built from circles centered at $(0,0), (3,0), (1.5, 0.4)$ with radii of 1,1, and 1.5 respectively. The first two half circles lie in the upper half plane, while the third circle lies in the lower half plane. The circles are then embedded into $\mathbb{R}^{100}$ by setting the remaining 98 coordinates to zero. Finally, Gaussian noise with mean zero and standard deviation 0.14 is added to each of the 100 coordinates.

We construct the dataset by sampling 500 points from each of the three circles, for a total of 1500 points. We construct the weight matrix using the 15 nearest neighbors with local scaling by the 7^{th} nearest neighbor.

5.4.1.2 MNIST

MNIST is a database of 70,000 grayscale 28×28 pixel images of handwritten digits (0-9). Each of the digits is centered and size normalized. The MNIST dataset may be downloaded at http://yann.lecun.com/exdb/mnist/ The data set is separated into 60,000 training images and 10,000 test images. We combine them to create a single set of 70,000 images to test against. We perform no preprocessing on the images. We construct the weight matrix using the 15 nearest neighbors with local scaling based on the 7^{th} nearest neighbor.

5.4.1.3 Opt-Digits

Opt-Digits is a database of 5620 handwritten digits [51]. The data is recorded as an 8×8 integer matrix, where each element is between 0 and 16. The dataset may be downloaded at https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits. We construct the weight matrix using the 15 nearest neighbors and local scaling by the 7^{th} nearest neighbor.

5.4.1.4 COIL

The Columbia Object Image Library (COIL-100) is a database of 128×128 pixel color images of 100 different objects photographed at various different angles [69]. In [70] the authors processed the COIL images to create a more difficult benchmark set. The red channel of each image is downsampled to 16×16 pixels by averaging over blocks of 8×8 pixels. The images are then further distorted and downsampled to create 241 dimensional feature vectors. Then 24 of the objects are randomly selected and randomly partitioned into 6 different classes. Discarding 38 images from each class leaves 250 images per class for a total of 1500 points. This benchmark set may be found at http://olivier.chapelle.cc/ssl-book/benchmarks.html. We construct the weight matrix using the 4 nearest neighbors and local scaling by the 4^{th} nearest neighbor.

5.4.2 Experimental results using GHCMBO and GHCMBOS

In this subsection, we compare GHCMBO and GHCMBOS (without temperature) to previous graph MBO schemes (MBO eigenvectors [33], HKPR1/2 MBO [60]) and the total variation based convex method (TVRF [79]). The results reported for the other methods are taken from their respective papers (hence the blank entries in certain columns).

For GHCMBO we choose the diffusion matrix to be $A = W^2$, and in GHCMBOS we take $A = W$. In all experiments we set $\tau = 0.1$ in the fidelity term. Non-fidelity nodes $x \in \mathcal{V} \setminus F$ are initialized by assigning x to phase $i(x) = \arg\min_{1 \le j \le N} f_j(x)$. We run the algorithms until a fixed point is reached. On average, convergence requires between 10 to 30 iterations depending on the size of $\mathcal{V}$ and F. The reported results are averaged over 100 trials. In each trial, the fidelity set F is chosen at random, and the size of each class in F is allowed to be random. Results are in Tables 5.2-5.6.

Timing comparisons show that GHCMBO and GHCMBOS are between two to five orders of magnitude faster than other state of the art variational methods. In addition, our methods are comparable or superior to the compared methods. On the MNIST and COIL datasets, we outperform all other methods. We also note that based on the results in Table 5.4, our choice of linear fidelity term appears to better capture the structure of the data as compared to the region force term in [79].

Table 5.2: Three Moons

Method/ Fidelity %	1.66%	3.33%	5%	Timing (ms)
TVRF [79]	96.4%	98.2%	98.6%	–
MBO eigenvectors [33]	–	–	**99.12**%	344
GHCMBO	97.45%	98.61%	98.94%	4.1
GHCMBOS	**97.81**%	**98.93**%	99.08%	3.1

Table 5.3: MNIST

Method/ Fidelity %	0.25%	0.5%	1%	3.6%	Timing (s)
TVRF [79]	94.6%	96.6%	96.8%	–	61
HKPR1 MBO [60]	–	–	–	97.52%	22.3
HKPR2 MBO [60]	–	–	–	97.48%	4,428
MBO eigenvectors [33]	–	–	–	96.91%	1,699
GHCMBO	**95.97**%	**96.81**%	**97.09**%	**97.54**%	0.30
GHCMBOS	92.91%	95.33%	96.32%	97.27%	0.17

Table 5.4: Comparing Fidelity terms on MNIST

Method/ Fidelity %	0.25%	0.5%	1%	Timing (s)
Fidelity only [79]	35.5%	52.3%	71.5%	0.4
Fidelity only (5.19)	**84.93%**	**88.61%**	**90.90%**	0.13

Table 5.5: Opt-Digits

Method/ Fidelity %	0.89%	1.78%	2.67%	Timing (ms)
TVRF [79]	**95.9%**	97.2%	**98.3%**	–
GHCMBO	95.68%	**97.63%**	98.10%	15.4
GHCMBOS	94.20%	96.30%	97.28%	11.0

Table 5.6: COIL

Method/ Fidelity %	3.33%	6.66%	10%	Timing (ms)
TVRF [79]	80.3%	90.0%	91.7%	–
MBO eigenvectors [33]	–	–	91.46%	220
HKPR1 MBO [60]	–	–	91.09%	1,000
HKPR2 MBO [60]	–	–	91.23%	92,000
GHCMBO	**83.01%**	92.24%	94.30%	1.00
GHCMBOS	82.96%	**92.30%**	**94.34%**	0.76

5.4.3 Experimental results using GHCMBOV

In this subsection, we test GHCMBOV on the MNIST, and Optdigits benchmark datasets. The algorithm is tested both with and without temperature and using several different volume bounds. We set the upper and lower bounds, $\boldsymbol{U}$ and $\boldsymbol{B}$ respectively, to be $B_i = V_i(1-x)$ and $U_i = V_i(1+x)$ where V_i is the ground truth volume of phase i and $x \in \{0, \frac{1}{10}, \ldots, \frac{4}{10}\}$. When temperature is used, we set $T = 0.1$. We choose $A = W^2$ for our diffusion matrix. We use the fidelity term only to initialize non-fidelity nodes (i.e. we set $\tau = 0$). The non-fidelity nodes $x \in \mathcal{V} \backslash F$ are assigned to phase $i(x) = \arg\min_{1 \leq j \leq N} f_j(x)$.

If there is no temperature, we run GHCMBOV until a fixed point is reached. On average, convergence requires between 10 to 30 iterations depending on the size of $\mathcal{V}$ and F. In our experiments with temperature, we run the algorithm for 100 iterations and extract the configuration with the lowest energy. The reported results are averaged over 100 trials. In each trial, the fidelity set F is chosen at random, and the size of each class in F is allowed

to be random. Timing results are in Table 5.7. Accuracy results are in Tables 5.8-5.9. We are able to get excellent accuracy results at extremely low fidelity percentages that are not feasible for the above variational methods [33, 60, 79].

Table 5.7: GHCMBOV Timing (in seconds)

Data Set	Bounds / Fid. %	1.0 1.0	1.2 0.8	1.4 0.6	no size constraints
MNIST	5%	**9.9920** / 0.5546	**7.5012** / 0.3393	**7.2017** / 0.2977	**7.1802** / 0.2793
	0.05%	**10.8273** / 2.6292	**9.3525** / 1.8574	**9.1032** / 1.2187	**8.0331** / 0.6370
OptDigits	20%	**0.7405** / 0.0219	**0.5889** / 0.0152	**0.5850** / 0.0141	**0.5838** / 0.0139
	0.4%	**0.8616** / 0.0352	**0.7319** / 0.0341	**0.7225** / 0.0336	**0.6266** / 0.0206

Bold= with temperature, not bold= without temperature

Table 5.8: GHCMBOV MNIST Results

Bounds Fid. %	1.0 1.0	1.1 0.9	1.2 0.8	1.3 0.7	1.4 0.6	no size constraints
0.05%	**94.837634** 91.007646	**93.894646** 89.828471	**93.172451** 88.095934	**91.484446** 87.124446	**89.661789** 85.866606	**83.490960** 82.620017
0.06%	**95.570360** 93.025889	**94.595109** 91.255817	**93.884229** 89.988214	**92.539709** 89.000709	**91.374931** 87.848183	**86.428977** 85.253046
0.075%	**96.424669** 94.654037	**95.831903** 93.305451	**94.928177** 91.951574	**94.020520** 90.992777	**92.950177** 90.534463	**90.722977** 89.396774
0.1%	**96.878680** 95.963954	**96.391851** 94.700349	**95.874583** 93.785543	**95.197703** 93.053509	**94.871051** 92.741034	**93.155503** 92.121651
0.15%	**97.196171** 96.557777	**96.766817** 95.674943	**96.526971** 95.247357	**96.114154** 95.035223	**95.942749** 94.906280	**95.681137** 94.830649
0.2%	**97.277606** 96.848260	**96.915549** 96.148926	**96.789714** 95.988211	**96.699503** 95.877506	**96.643309** 95.842009	**96.541497** 95.827191
0.5%	**97.378091** 97.155603	**97.215657** 96.902894	**97.197566** 96.894609	**97.195926** 96.889983	**97.191120** 96.882623	**97.187600** 96.875426
1.0%	**97.429246** 97.313697	**97.313629** 97.175394	**97.309011** 97.174094	**97.306183** 97.164149	**97.305103** 97.153377	**97.303709** 97.150551
2.5%	**97.542469** 97.511909	**97.477206** 97.414689	**97.469183** 97.424026	**97.463874** 97.423431	**97.461366** 97.423111	**97.455634** 97.422223

Bold= with temperature, not bold= without temperature

Table 5.9: GHCMBOV Optdigits Results.

Bounds Fid. %	1.0 1.0	1.1 0.9	1.2 0.8	1.3 0.7	1.4 0.6	no size constraints
0.4%	**93.043167** 86.865648	**92.380721** 86.798397	**91.702500** 86.163779	**91.060649** 85.574292	**89.963372** 85.096671	**85.293114** 83.384948
0.45%	**94.894706** 89.758893	**94.226059** 89.362007	**93.578479** 88.618025	**92.870979** 87.980644	**91.911868** 87.639002	**87.924751** 85.933278
0.5%	**95.964875** 91.762050	**95.175979** 91.055883	**94.664021** 90.390260	**93.842616** 89.870701	**93.055881** 89.310379	**89.758221** 87.980418
0.75%	**98.065649** 95.895059	**97.192660** 95.071735	**96.617260** 94.344114	**96.329875** 93.894988	**95.854929** 93.623212	**94.683372** 93.003228
1%	**98.385374** 97.105692	**97.567776** 96.239815	**97.140872** 95.692146	**96.914635** 95.401053	**96.751352** 95.256740	**96.331495** 95.037457
2%	**98.669635** 98.132231	**98.011646** 97.447865	**97.859217** 97.302979	**97.841753** 97.276091	**97.831984** 97.267961	**97.803523** 97.250568
3%	**98.740845** 98.424246	**98.238238** 97.925253	**98.216459** 97.865655	**98.186557** 97.856125	**98.178694** 97.851680	**98.173790** 97.846831

Bold= with temperature, not bold= without temperature

5.5 Conclusions

We have presented three MBO schemes, GHCMBO, GHCMBOS, and GHCMBOV for solving the SSL problem on a weighted graph. Our schemes are based on the graph heat content energy (GHC) and the theory developed in [27] and the previous chapters. We solve the SSL problem by minimizing an energy that combines GHC with a linear fidelity term based on graph geodesics, inspired by the region force in [79]. GHC depends on the choice of affinity matrix A, which induces a diffusion process on the graph. If A is PSD then GHCMBO and GHCMBOV decrease the energy at every step, while GHCMBOS

decreases the energy for an even more general class of diffusion matrices. Our approach considerably generalizes and simplifies previous SSL graph MBO schemes [33], [60]. The guarantee of energy dissipation and convergence to local minima is a new and important theoretical advance for SSL graph MBO schemes.

Experimental results on benchmark datasets show that our algorithms produce results with comparable or superior accuracy to other state-of-the-art methods [33], [60], [79]. GHCMBOV in particular, remains accurate and effective for fidelity percentages that are out of reach for other SSL methods. In addition, all of our schemes are faster than the current state of the art. GHCMBO and GHCMBOS, were nearly two orders of magnitude faster than [33], [60], and [79] on every dataset. Our algorithms are so fast because we are free to choose diffusions generated by extremely sparse matrices, and take very large step sizes through the configuration space.

CHAPTER 6

Conclusion

In this work we have derived and analyzed new algorithms and convolution kernels for simulating weighted curvature motion and solving minimal partition problems. The heart of our approach is the variational interpretation of the MBO scheme developed in [27] based on the heat content energy.

In Chapter 2, we derived new "single growth" versions of the MBO scheme that allowed us to guarantee unconditional stability for an extremely wide class of kernels. In particular, Algorithm 2.3 guarantees unconditional stability of the multi-phase scheme for essentially any network of kernels. This full level of generality is a first for threshold dynamics schemes. We also proved some convergence results. In the two-phase case, our single growth scheme is shown to converge to the viscosity solution of the curvature flow. While we cannot show convergence to the flow in the multi-phase case, we show that the heat content energy Gamma-converges to the correct limiting energy under certain kernel conditions. Our Gamma-convergence result gives us hope that the energy type convergence arguments of [54] can be extended to more general MBO schemes studied here. Indeed, the admissible kernels for our Gamma-convergence argument are the most general to date and include sign-changing kernels. Unfortunately, the allowed sign-changing kernels cannot induce non-zonoidal anisotropies. We hope to address this shortcoming in future research (although progress seems difficult).

In Chapter 3, we gave a decisive and constructive answer to the inverse problem of designing a kernel to induce a particular surface tension and mobility under threshold dynamics. In two dimensions, we give a completely explicit construction of a non-negative kernel to induce any target smooth and convex pair (σ, μ) of anisotropy and mobility. In three dimensions, not every pair can be induced by a positive kernel. We give a complete description of the necessary and sufficient conditions on the targets (σ, μ) for nonnegativity of the kernel, and an explicit construction (modulo inversion of a Radon transform) for admissible (σ, μ). In both cases, our kernel is smooth and compactly supported. In addition, for any target smooth, convex pair (σ, μ) in any dimension, we provide a completely

explicit Fourier domain kernel construction. The resulting kernel is Schwartz class and non-negative in the Fourier domain. The sum of this chapter fully answers an important and longstanding question in the threshold dynamics community.

In Chapter 4, we introduced auction dynamics, a volume preserving variant of the MBO scheme. Auction dynamics simulates multi-phase volume preserving or volume bounded curvature flow. Each step of the algorithm involves solving the assignment problem, which we efficiently approach using auction algorithms. The main benefit of our scheme is the simultaneous update of the configuration and the Lagrange multipliers corresponding to the volume constraints. This allows us to use the algorithm in situations where the interfacial boundaries may be extremely rough or irregular. We also show how to incorporate random fluctuations due to temperature in the spirit of Monte-Carlo algorithms.

In Chapter 5, we extend the previous MBO algorithms to the graphical setting using the graph heat content energy (GHC). GHC is closely related to the graph cut, thus the minimization of GHC allows us to solve SSL graph partitioning problems. Our resulting schemes minimize GHC at every step, guaranteeing unconditional stability and convergence to a fixed point. Crucially, our algorithms are compatible with nearly any graph diffusion process. This generality allows us to choose diffusions generated by extremely sparse matrices. As a result, our algorithm has a low computational complexity. The ability to work with sparse matrices and guarantee energy dissipation is a first for graph MBO schemes. Experimental results on benchmark machine learning datasets show that our methods are comparable or superior to other state-of-the-art variational methods while running faster. In particular, our basic algorithms, GHCMBO and GHCMBOS, are at least two orders of magnitude faster than other state-of-the-art variational SSL methods. Our volume preserving method, GHCMBOV, remains effective at low fidelity percentages that are out of reach for competing state-of-the-art variational SSL methods.

BIBLIOGRAPHY

[1] G. Alberti and G. Bellettini. A non-local anisotropic model for phase transitions: asymptotic behavior of rescaled energies. *European J. Appl. Math.*, 9:261–284, 1998.

[2] F. Almgren, J. E. Taylor, and L.-H. Wang. Curvature-driven flows: a variational approach. *SIAM Journal on Control and Optimization*, 31(2):387–438, 1993.

[3] L. Ambrosio and A. Braides. Functionals defined on partitions in sets of finite perimeter II: semicontinuity, relaxation, and homogenization. *J. Math. Pures et Appl.*, 69:307–333., 1990.

[4] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM J. Numer. Anal.*, 32:484–500, 1995.

[5] G. Barles and P. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.

[6] J. W. Barrett, H. Garcke, and R. Nürnberg. Parametric approximation of surface clusters driven by isotropic and anisotropic surface energies. *Interfaces and Free Boundaries*, 12:187–234, 2010.

[7] G. Bellettini and M. Paolini. Anisotropic motion by mean curvature in the context of Finsler geometry. *Hokkaido Mathematical Journal*, 25:537–566, 1996.

[8] Andrea L. Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling and Simulation*, 10(3):1090–1118, 2012.

[9] Dimitri Bertsekas. A distributed algorithm for the assignment problem. Technical report, MIT, May 1979.

[10] Dimitri Bertsekas. *Linear network optimization*. MIT Press, 1991.

[11] Dimitri Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[12] Dimitri Bertsekas and David Castanon. The auction algorithm for the transportation problem. *Annals of Operations Research*, 20:67–69, 1989.

[13] Dimitri Bertsekas, David Castanon, and Haralampos Tsaknakis. Reverse auction and the solution of asymmetric assignment problems. *SIAM J. on Optimization*, 3:268–299, 1993.

[14] E. D. Bolker. A class of convex bodies. *Transactions of the American Mathematical Society*, 145:323–345, 1969.

[15] E. Bonnetier, E. Bretin, and A. Chambolle. Consistency result for a non-monotone scheme for anisotropic mean curvature flow. *Interfaces and Free Boundaries*, 14(1):1–35, 2012.

[16] X. Bresson, T. Chan, X. Tai, and A. Szlam. Multiclass transductive learning based on L1 relaxations of cheeger cut and Mumford-Shah-Potts model. *Journal of Mathematical Imaging and Vision*, 49(1):191–201, August 2013.

[17] L. Bronsard and B. Wetton. A numerical method for tracking curve networks moving with curvature motion. *Journal of Computational Physics*, 120(1):66–87, 1995.

[18] H. Busemann and C. M. Petty. Problems on convex bodies. *Mathematica Scandinavica*, 4:88–94, 1956.

[19] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in Analysis*, pages 195–199. 1970.

[20] Y.-G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *Journal of Differential Geometry*, 33:749–786, 1991.

[21] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 2005.

[22] E. De Giorgi. New problems on minimizing movements. In C. Baiocchi and J. L. Lions, editors, *Boundary value problems for PDE and applications*, pages 81–98. Masson, 1993.

[23] A. di Carlo, M. E. Gurtin, and P. Podio-Guidugli. A regularized equation for anisotropic motion-by-curvature. *SIAM Journal on Applied Mathematics*, 52(4):1111–1119, 1992.

[24] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[25] M. Elsey and S. Esedoḡlu. Threshold dynamics for anisotropic surface energies. *AMS Mathematics of Computation*, 2017.

[26] S. Esedoḡlu and M. Jacobs. Convolution kernels and stability of threshold dynamics methods. Technical report, University of Michigan, August 2016. Under review.

[27] S. Esedoḡlu and F. Otto. Threshold dynamics for networks with arbitrary surface tensions. *Communications on Pure and Applied Mathematics*, 68(5):808–864, 2015.

[28] S. Esedoḡlu and Y.-H. Tsai. Threshold dynamics for the piecewise constant Mumford-Shah functional. *Journal of Computational Physics*, 211(1):367–384, 2006.

[29] Selim Esedoglu, Matt Jacobs, and Pengo Zhang. Kernels with prescribed surface tension and mobility for threshold dynamics schemes. *Journal of Computational Physics*, 337:62–83, 15 May 2017.

[30] L. C. Evans. Convergence of an algorithm for mean curvature motion. *Indiana University Mathematics Journal*, 42:553–557, 1993.

[31] L. C. Evans and J. Spruck. Motion of level sets by mean curvature. I. *Journal of Differential Geometry*, 33:635–681, 1991.

[32] P. Funk. Über flächen mit lauter geschlossenen geodätischen linien. *Math. Ann.*, 74(2):278–300 (German), 1913.

[33] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1600–1613, 2014.

[34] R. J. Gardner. Intersection bodeis and the Busemann-Petty problem. *Transactions of the American Mathematical Society*, 342(1):435–445, 1994.

[35] R. J. Gardner. A positive answer to the Busemann-Petty problem in three dimensions. *Annals of Mathematics*, 140:435–447, 1994.

[36] A.V. Goldberg. Solving minimum-cost flow problems by successive approximation. In *STOC 87*, November 1986.

[37] A.V. Goldberg. Efficient graph algorithms for sequential and parallel computer. Technical report, Laboratory for Computer Science, M.I.T., 1987.

[38] A.V. Goldberg and R.E. Tarjan. Solving minimum cost flow problems by successive approximation. In *Proc. 19th ACM STOC*, May 1987.

[39] Paul Goodey and Wolfgang Weil. Centrally symmetric convex bodies and the spherical Radon transform. *Journal of Differential Geometry*, 35:675–688, 1992.

[40] Matt Grayson. Shortening embedded curves. *The Annals of Mathematics*, 129(1):71–111, 1989.

[41] Thomas Hales. The honeycomb conjecture. *Discrete and Computational Geometry*, 25(1):1–22, 2001.

[42] M. Hein and S. Setzer. Beyond spectral clustering - tight relaxations of balanced graph cuts. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, 2011.

[43] S. Helgason. *The Radon Transform*, volume 5 of *Progress in Mathematics*. Birkhauser, Boston, MA, 1999.

[44] C. Herring. *The Physics of Powder Metallurgy*, chapter Surface tension as a motivation for sintering, pages 143–179. McGraw Hill, 1951.

[45] Tony Hoare. Algorithm 65: Find. *Communications of the ACM*, 4(7):321–322, 1961.

[46] E. A. Holm, G. N. Hassold, and M. A. Miodownik. Dimensional effects on anisotropic grain growth. In G. Gottstein and D. Molodov, editors, *Recrystallization and Grain Growth*, pages 239–244, RWTH Aachen, Germany, August 2001. First Joint International Conference on Recrystallization and Grain Growth, Springer-Verlag.

[47] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Removing the stiffness from interfacial flows with surface tension. *Journal of Computational Physics*, 114(2):312–338, 1994.

[48] H. Ishii, G. E. Pires, and P. E. Souganidis. Threshold dynamics type approximation schemes for propagating fronts. *Journal of the Mathematical Society of Japan*, 51:267–308, 1999.

[49] Matt Jacobs. A fast MBO scheme for multiclass data segmentation. In *Scale Space and Variational Methods in Computer Vision*, 2017.

[50] Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoḡlu. Auction dynamics: A volume preserving MBO scheme. Technical report, In preperation, 2017.

[51] C Kaynak. Methods of combining multiple classifiers and their applications to handwritten digit recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, 1995.

[52] D. Kinderlehrer, I. Livshitz, and S. Taasan. A variational approach to modeling and simulation of grain growth. *SIAM Journal on Scientific Computing*, 28(5):1694–1715, 2006.

[53] Harold Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2, 1955.

[54] T. Laux and F. Otto. Convergence of the thresholding scheme for multi-phase mean-curvature flow. Technical report, Max Planck Institute for Mathematics in the Sciences, 2015.

[55] S. Luckhaus and T. Sturzenhecker. Implicit time discretization for the mean curvature flow equation. *Calculus of Variations and Partial Differential Equations*, 3(2):253–271, 1995.

[56] E. Lutwak. Intersection bodies and dual mixed volumes. *Advances in Mathematics*, 71:232–261, 1988.

[57] J. Malik and J. Shi. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[58] P. Mascarenhas. Diffusion generated motion by mean curvature. CAM Report 92-33, UCLA, July 1992. (URL = http://www.math.ucla.edu/applied/cam/index.html).

[59] Peter McMullen and Egon Schulte. *Abstract Regular Polytopes*. Macmillan, 2002.

[60] E. Merkurjev, A. Bertozzi, and F. Chung. A semi-supervised heat kernel pagerank MBO algorithm for data classification. *Submitted*, 2016.

[61] Ekaterina Merkurjev, Justin Sunu, and Andrea L. Bertozzi. Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In *IEEE International Conference on Image Processing*, October 2014.

[62] B. Merriman, J. K. Bence, and S. Osher. Motion of multiple junctions: a level set approach. *Journal of Computational Physics*, 112(2):334–363, 1994.

[63] B. Merriman, J. K. Bence, and S. J. Osher. Diffusion generated motion by mean curvature. In J. Taylor, editor, *Proceedings of the Computational Crystal Growers Workshop*, pages 73–83. AMS, 1992.

[64] M. Miranda, D. Pallara, F. Paronetto, and M. Preunkert. Short-time heat flow and functions of bounded variation in $\mathbb{R}^N$. *Ann. Fac. Sci. Toulouse, Mathematiques*, 16(1):125–145, 2007.

[65] F. Morgan. Lowersemicontinuity of energy clusters. *Proceedings of the Royal Society of Edinburgh*, 127A:819–822, 1997.

[66] W. W. Mullins. Two dimensional motion of idealized grain boundaries. *J. Appl. Phys.*, 27:900–904, 1956.

[67] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[68] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1), 1957.

[69] S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical report, Columbia University, 1996.

[70] Alexander Zien Olivier Chapelle, Bernhard Scholkopf. *Semi-Supervised Learning*. The MIT Press, 2006.

[71] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[72] S. J. Ruuth. *Efficient algorithms for diffusion-generated motion by mean curvature*. PhD thesis, The University of British Columbia, 1996.

[73] S. J. Ruuth. A diffusion generated approach to multiphase motion. *Journal of Computational Physics*, 145:166–192, 1998.

[74] S. J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *Journal of Computational Physics*, 144:603–625, 1998.

[75] S. J. Ruuth and B. Merriman. Convolution generated motion and generalized Huygens' principles for interface motion. *SIAM Journal on Applied Mathematics*, 60:868–890, 2000.

[76] S. J. Ruuth and B. Wetton. A simple scheme for volume-preserving motion by mean curvature. *Journal of Scientific Computing*, 19(1):373–384, 2003.

[77] D. Weaire and R. Phelan. A counter-example to kelvin's conjecture on minimal surfaces. *Philosophical Magazine Letters*, 69:107–110, 1994.

[78] X. Xu, D. Wang, and X. Wang. An efficient threshold dynamics method for wetting on rough surfaces. *arXiv:1602.04688*, February 2016.

[79] K. Yin, Xue-Cheng Tai, and S. J. Osher. An effective region force for some variational models for learning and clustering. Technical report, UCLA, 2016.

[80] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Ninth IEEE International Coference on Computer Vision, 2003*, volume 1, pages 313–319, October 2003.

[81] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *Advances in neural information processing systems*, 2004.